

RDBS Oracle v.7

Relační databázový systém Oracle je systém typu klient-server. Serverová část běží na stroji „sts“, klienti na jednotlivých počítačích v učebnách.

Se systémem je možné pracovat dvěma způsoby. Program „SQL*Plus“ je interpret SQL příkazů. Takto je možné s Oraclem pracovat v řádkovém interaktivním režimu zadáváním jednotlivých SQL příkazů. Tento způsob je vhodný zejména pro ladění a kontrolu obsahu databázových tabulek. Druhou možností je použít vývojové prostředí „Developer 2000“. V tomto prostředí je možné vizuálním grafickým způsobem vytvářet uživatelské aplikace, které pracují s databází Oraclu.

V následujícím textu jsou popsány základní programové prostředky a systémy pro práci a vývoj aplikací v systému Oracle. Jsou popsány v tom pořadí a tím způsobem, jak je bude uživatel používat při tvorbě vlastní aplikace.

1 Security Manager

Každý uživatel systému Oracle má zřízen svůj uživatelský účet identifikovaný jménem uživatele a chráněn heslem. Při zahájení práce s libovolným programovým prostředkem je nutné se nejprve (do Oraclu) přihlásit. Toto přihlašování do Oraclu nemá nic společného s přihlašováním do Novellu nebo Unixu.

Program „Security Manager“ slouží pro správu uživatelských účtů. Nalézá se ve složce „Start“ - „Programs“ - „Oracle Enterprise Manager“ - „Security Manager“.

V tomto programu je možné měnit svoje přístupové heslo do systému Oracle a prohlédnout si přidělená přístupová práva.

Po spuštění je nutné se připojit do databáze. Zadáme své jméno, heslo a databázi, do které se připojujeme. Jméno databáze je vždy „stud“.

Zobrazí se dvě okna vedle sebe. V levém okně rozbalíte složku „Users“ a najdete se (najdete složku s Vaším přihlašovacím jménem). Klepnutím levým tlačítkem myši na jméno nebo postavičku vedle něj se v pravém okně zobrazí informace o daném uživateli.

Ve složce „General“ v sekci „Authentication“ je možné zadat a potvrdit nové přístupové heslo uživatele.

Rozbalením Vaší složky v levém okně si můžete prohlédnout přístupová práva, která Vám byla přidělena.

Poznámka :

Nenechte se zlákat množstvím dalších funkcí a informací. „Security Manager“ Vám slouží pouze pro změnu hesla. Nemáte přístupová práva na změnu jakékoliv jiné informace, resp. na uložení těchto změn.

2 Database Designer

Program „Database Designer“ slouží pro návrh datového modelu aplikace, tedy struktury a vztahů databázových tabulek. Nalézá se ve složce „Start“ - „Programs“ - „Database Designer“ - „Database Designer“. Umožňuje grafický návrh databázových tabulek způsobem podobným tvorbě E-R diagramu. Podle výsledného návrhu je poté vygenerován SQL skript. Ten po svém spuštění (interpretem SQL příkazů, např. „SQL*Plus“) provede vlastní vytvoření databázových tabulek, indexů, pohledů a dalších komponent datového modelu aplikace.

2.1 Popis Database Designeru

Jednotlivé funkce v menu „File“, „Edit“, „View“, „Utilities“, „Window“ a „Help“ jsou zřejmě již ze svých názvů. V menu „Tools“ jsou uloženy následující funkce. „Generation Wizard“ slouží pro vygenerování SQL skriptu z hotového datového modelu. „Reverse Engineer Wizard“ provádí opačný postup. Z již existujících databázových tabulek vyextrahuje jejich popis a vytvoří z něj popis struktury tabulek. Funkce „SQL*Plus“ by měla spouštět interpret SQL příkazů.

Ikony v prvním řádku pod menu odpovídají jednotlivým nejčastěji používaným funkcím z menu.

Ikony v druhém řádku označují funkce pro editaci diagramu. Funkce „Select“ označuje již existující objekt v diagramu. Funkce „Table“, „View“, „Mandatory Foreign Key“, „Optional Foreign Key“ a „Primary Key“ slouží pro vytvoření odpovídajícího objektu v diagramu (datovém modelu). Zbývající funkce slouží pro změnu barev.

2.2 Postup při tvorbě datového modelu

Vytvoření databázové tabulky

Kliknete levým tlačítkem myši na ikonu „Table“. Na tom místě okna diagramu, kde chcete mít novou tabulku, kliknete myši znovu. Pokud držíte tlačítko myši stisknuté, můžete změnit velikost symbolu nové tabulky.

Po dvojkliku levým tlačítkem myši na symbolu nové tabulky se otevře dialogové okno, ve kterém se zadávají parametry nové tabulky.

Ve složce „Table“ se zadávají položky „Name“ (jméno tabulky), případně „Alias“ (alias jméno tabulky), „Display Title“ (nadpis zobrazovaný místo jména tabulky ve výpisech) a případně „Comment“ (doplňující poznámka). Ve složce „Column Defn“ se definují jednotlivé sloupce databázové tabulky. Pro každý sloupec se vyplňují položky „Column Name“ (jméno sloupce), „Datatype“ (typ sloupce. Zde vřele doporučuji používat POUZE datové typy VarChar2, Number a Date. Ušetříte si tím spoustu problémů), „Max Len“ (délka sloupce, tedy maximální počet znaků nebo číslic ve sloupci), „Dec Pl“ (počet desetinných míst) a „Default Value“ (implicitní hodnota).

Ve složce „Col Display“ se dá případně zadat způsob, jakým se jednotlivé sloupce tabulky budou zobrazovat.

Ve složce „Constraints“ se zadávají integritní omezení kladená na tabulku. V podsložce „Primary“ se definuje primární klíč tabulky. Zadávají se položky „Constraint Name“ (název integritního omezení), může se zadat položka „Error Message“ (chybové hlášení zobrazené při porušení integritního omezení), v dolním obdélníku „Seq“ (jedinečné pořadové číslo) a „Column Name“ (jméno sloupce tabulky, který tvoří primární klíč). V podsložce „Foreign“ se stejným způsobem definují cizí klíče tabulky. Vyplňuje se navíc položka „Join Table“ označující tabulku, která je zadávaným cizím klíčem připojena k aktuální tabulce a tvoří s ní relaci. V podsložce „Unique“ se podobným způsobem zadávají sloupce, které mají mít jedinečné hodnoty a v podsložce „Check“ obecná integritní omezení.

Ve složce „Validation“ se zadávají validační podmínky, které určují přípustné hodnoty jednotlivých sloupců tabulky.

Ve složce „Index“ se vytvářejí indexy pro danou tabulku. Zadávají se položky „Index Name“ (název vytvářeného indexu), v dolním obdélníku „Seq“ (jedinečné pořadové číslo) a „Column Name“ (jméno sloupce tabulky, který tvoří index).

Primární klíč tabulky je možné označit také pomocí ikony „Primary Key“.

Poznámka :

Symbol tabulky může být umístěn na libovolné místo diagramu. Umístění je dáno pouze přehledností celého návrhu.

Vytvoření pohledu

Kliknete levým tlačítkem myši na ikonu „View“. Na tom místě okna diagramu, kde chcete mít nový pohled, kliknete myši znovu. Pokud držíte tlačítko myši stisknuté, můžete změnit velikost symbolu nového pohledu.

Po dvojkliku levým tlačítkem myši na symbolu nového pohledu se otevře dialogové okno, ve kterém se zadávají parametry nového pohledu.

Ve složce „View“ se zadávají položky „Name“ (název pohledu), „Alias“ (alias jméno pohledu), „Display Title“ (nadpis zobrazovaný místo jména pohledu ve výpisech) a případně „Comment“ (komentář).

Ve složkách „Column Defn“, „Col Display“, „Constraints“ a „Validation“ se stejně jako při vytváření tabulky zadávají položky (sloupce) pohledu, způsob jejich zobrazení, integritní omezení a validační podmínky.

Ve složce „Base Relations“ se zadávají tabulky nebo jiné pohledy, jejichž sloupce budou zobrazeny v zadávaném pohledu. Pro každou takovou tabulku nebo pohled se vyplňují položky „Seq No“ (jedinečné pořadové číslo), „Table/View“ (jméno tabulky či pohledu) a případně „Alias“ (alias jméno tabulky či pohledu).

Ve složce „Base Cols“ se zadává, které položky (sloupce) zadávaného pohledu odpovídají kterým sloupcům tabulek či jiných pohledů spojených v zadávaném pohledu. Vyplňují se položky „Column Name“ (jméno sloupce zadávaného pohledu), „Seq“ (jedinečné pořadové číslo), „Base Column“ (jméno sloupce databázové tabulky nebo jiného pohledu, který je zobrazen v zadávaném pohledu. Zadává se ve tvaru „jméno_tabulky.jméno_sloupce“), „Datatype“ (typ sloupce), „Max Len“ (velikost sloupce), „Dec Pl“ (počet desetinných míst sloupce) a „Expression“ (výraz udávající hodnotu sloupce zadávaného pohledu).

Vytvoření relací

Výše popsaným způsobem se vytvoří potřebné databázové tabulky a pohledy. Nyní je možné tabulky a pohledy spojit do relací. Klikneme na ikonu „Mandatory Foreign Key“, resp. „Optional Foreign Key“ pro vytvoření požadovaného typu relace. Nyní postupně klikneme na tabulky, které mají spojeny relací a stane se tak.

Uložení datového modelu do souboru

Pomocí funkce „Save“ nebo „Save As“ zde vytvořený datový model uloží do souboru na disku.

Vygenerování skriptu

Kliknutím na příslušnou ikonu nebo v menu volbou funkce „Tools“ - „Generation Wizard“ se spustí generování SQL skriptu. Generovaný SQL skript obsahuje SQL příkazy, které vytvoří databázové tabulky podle hotového datového modelu. Skript se vygeneruje do textového souboru. Zvolte tedy volby „File“ a zadejte výstupní soubor. Položka „Database type“ musí být nastavena na hodnotu „Oracle7“. Je-li zaškrtnuto pole „View generated file“, skript je po vygenerování zobrazen na obrazovce. Stiskem tlačítka „Next“ se přesuneme do dalšího okna. Zde necháme volbu „All types at once“ a znovu stiskneme tlačítko „Next“. Nyní je ještě možné vybrat, které objekty skutečně budou generovány. Ponecháme beze změny a stiskem tlačítka „Finish“ spustíme generování skriptu.

Vygenerovaný skript spustíme v programu „SQL Worksheet“ nebo „SQL*Plus“ (podrobnější popis viz. dále), aby podle něj byly tabulky a pohledy skutečně vytvořeny.

3 SQL Worksheet

Program „SQL Worksheet“ je interpret SQL příkazů. Nachází se ve složce „Start“ - „Programs“ - „Oracle Enterprise Manager“ - „SQL Worksheet“.

Po spuštění je nutné se připojit do databáze. Zadáme své jméno, heslo a databázi, do které se připojujeme. Jméno databáze je vždy „stud“.

Po spuštění se zobrazí dvě okna SQL Worksheetu. V dolním se zapisují SQL příkazy a v horním je potom zobrazován průběh jejich provádění, stavová a chybová hlášení.

Chceme-li zadávat SQL příkazy ručně, píšeme je do dolního okna, případně oddělené středníky. Po zapsání SQL příkazu nebo příkazů, které chceme nechat provést je nutno zvolit funkci „Run Script“ z menu „Worksheet“, která zapsané příkazy spustí a ty se provedou.

Posloupnost SQL příkazů, tedy SQL skript, můžeme mít předem vytvořený a uložený v souboru, např. vygenerovaný „Database Designerem“. Pokud chceme takový SQL skript nechat provést, je třeba ho nejprve načíst pomocí funkce „Open“ v menu „File“. Načtený SQL skript opět spustíme funkcí „Run Script“ z menu „Worksheet“.

4 SQL*Plus

Program „SQL*Plus“ je další interpret SQL příkazů. Nachází se ve složce „Start“ - „Programs“ - „Oracle for Windows 95/NT“ - „SQL Plus 3.3“.

Po spuštění je nutné se připojit do databáze. Zadáme své jméno, heslo a databázi, do které se připojujeme. Jméno databáze je vždy „UIVTstud“, a to včetně velkých a malých písmen!

Objeví se prompt „SQL>“ a je možné zadávat SQL příkazy ukončené středníkem.

Pro spuštění SQL skriptu je nejjednodušší skript zobrazit v editoru (NotePad) a přes clipboard ho přenést do „SQL*Plus“, kde se automaticky spustí.

5 Schema Manager

Program „Schema Manager“ slouží pro správu jednotlivých komponent Oracleovské databáze, jako jsou tabulky, pohledy, snímky databáze, uložené procedury a databázové trigry, apod. Nachází se ve složce „Start“ - „Programs“ - „Oracle Enterprise Manager“ - „Schema Manager“.

Program „Schema Manager“ je vhodné používat pro drobné zásahy a opravy struktury databázových tabulek, případně indexů. Tedy optimální postup je pomocí „Database Designeru“ navrhnout a vytvořit databázové tabulky a případné drobné změny (které by však při dobrém prvotním návrhu neměly nastat) realizovat pomocí „Schema Manageru“. „Schema Manager“ je možné také použít pro přímé „ruční“ vytvoření tabulek, pokud nechceme použít „Database Designer“.

Dále je možné „Schema Manager“ použít pro vytváření, editování a případně rušení databázových trigrů a uložených procedur.

5.1 Popis Schema Manageru

V menu „File“ je zajímavá pouze funkce „Exit“. Jednotlivé funkce v menu „View“ jsou zřejmé již ze svých názvů. V menu „Object“ jsou funkce pro vytváření, modifikování a rušení objektů (tabulky, indexy, apod.) a pro přidělování práv přístupu k těmto objektům.

Ikony v prvním řádku pod menu odpovídají jednotlivým nejčastěji používaným funkcím z menu.

V levém okně je zobrazen hierarchický seznam jednotlivých typů databázových objektů, jejich instancí a vlastností. Rozbaluje se podobně jako seznam adresářů ve Windows Exploreru. Kliknutím na daný objekt se v pravém okně zobrazí dialogové okno se seznamem vlastností zvoleného objektu.

Po spuštění „Schema Manageru“ je nutné se připojit do databáze. Zadáme své jméno, heslo a databázi, do které se připojujeme. Jméno databáze je vždy „UIVTstud“, a to včetně velkých a malých písmen!

Dále je pak možno vytvářet, modifikovat nebo rušit databázové tabulky, indexy, databázové trigry nebo uložené procedury.

5.2 Databázová tabulka

Pokud chceme vytvářet novou tabulku, klikneme na ikonu „Create“ nebo zvolíme stejnou funkci z menu „Object“ a zvolíme typ vytvářeného objektu

„Table“. Pokud chceme tabulku opravovat, najdeme tuto tabulku v levém okně „Schema Manageru“ a klikneme na její název.

V zobrazeném dialogovém okně je možné zadat nebo opravit následující údaje.

Ve složce „General“ se zadávají položky „Name“ (jméno databázové tabulky) a „Schema“ (jméno vlastníka tabulky). V sekci „Columns“ se zadávají údaje o jednotlivých sloupcích tabulky. Tlačítka „New“, „Edit“ a „Remove“ slouží pro přidávání, editování a rušení sloupců. Pro každý sloupec tabulky se zadávají tyto údaje: „Name“ (jméno sloupce), „Datatype“ (typ sloupce. Doporučuji používat POUZE typy VarChar2, Number a Date), „Length“ (délka sloupce, tedy maximální počet znaků nebo číslic ve sloupci), „Precision“ (počet desetinných míst sloupce), „Default“ (implicitní hodnota sloupce) a „No Nulls“ (je-li zakázána prázdná hodnota sloupce).

Ve složce „Constraints“ se zadávají integritní a jiná omezení kladená na databázovou tabulku. Tlačítka „New“ a „Remove“ slouží pro vytvoření nového, resp. zrušení starého omezení. Zaškrťovací pole „Disable“ vyřazuje integritní omezení z činnosti. V sekci „Columns and Constraints“ je možné pro každý sloupec zadat, zda je primární nebo cizí klíč, zda má mít jedinečné hodnoty nebo zda na něj není kladeno žádné omezení. Do položky „Check“ je možné zadat obecné integritní omezení.

5.3 Index

Pokud chceme vytvářet nový index, klikneme na ikonu „Create“ nebo zvolíme stejnou funkci z menu „Object“ a zvolíme typ vytvářeného objektu „Index“. Pokud chceme index opravovat, najdeme tento index v levém okně „Schema Manageru“ a klikneme na jeho název.

V zobrazeném dialogovém okně je možné zadat nebo opravit následující údaje.

Ve složce „General“ se zadávají položky „Name“ (jméno indexu) a „Schema“ (jméno vlastníka indexu). V sekci „Index on“ se nastaví pole „Index on“ na hodnotu „Table“ (vytváříme index na tabulku) a hodnoty položek „Schema“ a „Table“ v této sekci nastavíme na vlastníka a tabulku, ke které vytváříme index. Potom zvolené sloupce tabulky kopírujeme z okna „Columns in Table“ do okna „Columns in Index“ v tom pořadí, v jakém budou tvořit indexovací výraz. V sekci „Options“ se volí další parametry indexu, např. zda index bude obsahovat jedinečné hodnoty.

5.4 Databázový trigr

Databázový trigr je programový kód, který se spustí výskytem nějaké události. Trigr se tedy nevolá jako procedura, ale jakmile v DATABÁZI (tedy ne ve formuláři nebo aplikaci) dojde k definované události, trigr se „sám“ spustí. Databázový trigr tedy může spustit např. vložení věty do tabulky v databázi, ale ne stisk tlačítka ve formuláři.

Databázový trigr je možné použít pro rozdělování výpočetní zátěže mezi klienta a server. Na rozdíl od aplikace, která běží na klientském počítači, je totiž vykonáván přímo na serveru. Hlavní oblast jeho použití je však hlídání referenční integrity dat v databázi.

Referenční integritu dat je možné zajistit dvěma způsoby. Prvním z nich je definice omezení databázové tabulky, zejména definice primárních a cizích klíčů. Oracle pak sám zajistí, že operace, která tato integritní omezení porušuje (např. vložení věty s duplicitním primárním klíčem, rušení věty, na kterou jsou v relaci navázány jiné věty z jiné tabulky, apod.), nebude provedena.

Druhým způsobem je použití databázového trigru, který podezřelou operaci zachytí a může na ni odpovídajícím způsobem zareagovat. Typicky se databázový trigr používá pro hlídání referenční integrity cizího klíče v relaci mezi tabulkami. V takovém případě může databázový trigr např. při rušení věty zrušit také všechny navazující věty v relaci svázané tabulce (tzv. kaskádní rušení) nebo při změně hodnoty atributu změnit odpovídajícím způsobem i hodnoty cizích klíčů v odpovídajících větách relací svázané tabulky, apod. Podrobněji viz. níže uvedený příklad použití.

Postup při vytváření nebo modifikaci trigrů

Pokud chceme vytvářet nový databázový trigr, klikneme na ikonu „Create“ nebo zvolíme stejnou funkci z menu „Object“ a zvolíme typ vytvářeného objektu „Trigger“. Pokud chceme trigr opravovat, najdeme ho v levém okně „Schema Manageru“ a klikneme na jeho název.

Po zadání všech parametrů nebo po provedení všech změn potvrdíme vytvoření trigru nebo uložení změn stiskem tlačítka „Apply“.

Popis databázového trigru

V zobrazeném dialogovém okně je možné zadat nebo opravit následující údaje.

Ve složce „General“ se zadávají základní údaje o trigru, zejména v položkách „Trigger“ (jméno trigru), „Schema“ (majitel trigru, tedy Vy). V sekci „Trigger On“ se zadává databázová tabulka, kterou bude trigr hlídat a na jejíž změny bude reagovat. Zadává se jménem vlastníka tabulky v položce „Schema“ a jménem tabulky v položce „Table“. A konečně v položce „Trigger Body“ se zapisují jednotlivé příkazy, které má trigr vykonat po svém spuštění. Je zde možné použít s jistými omezeními příkazy jazyka SQL i PL/SQL.

Ve složce „Timing“ se určuje kdy se trigr spustí, tedy na jaké události bude reagovat. V sekci „Triggering Statement“ se specifikuje druh události, na kterou má trigr reagovat. Trigr se může spustit buďto před (zapnutím přepínacího tlačítka „Before“) nebo po (zapnutím přepínacího tlačítka „After“) provedení specifikované akce. Touto akcí může být operace vložení věty do tabulky (zaškrtnutím boxu „Insert“), zrušení věty tabulky (zaškrtnutím boxu „Delete“) nebo modifikace věty tabulky (zaškrtnutím boxu „Update of Columns“). Pokud má trigr reagovat na modifikaci (update) věty tabulky, je třeba ještě zadat sloupce tabulky, na než má trigr reagovat, tedy jejichž modifikace trigr spustí. Je samozřejmě možné vytvořit trigr, který reaguje navíc druhů událostí, např. na INSERT a DELETE, a který je spuštěn libovolnou ze zaškrtnutých událostí.

Trigr může na výskyt zadané události reagovat dvěma způsoby. Buďto se spustí jen jednou před nebo po provedení dané akce, např. před provedením příkazu UPDATE, který může modifikovat i několik řádků tabulky. Nebo se trigr spustí několikrát před nebo po provedení akce s každým řádkem tabulky, který je vykonávaným SQL příkazem dotčen, např. před modifikací každého řádku tabulky, který vyhovuje podmínce WHERE příkazu UPDATE. O tom, která z těchto variant bude pro trigr platit, rozhoduje zaškrťovací box „Trigger for Each Row“. Je-li zaškrtnut, spustí se trigr pro každý řádek tabulky dotčený vykonávaným SQL příkazem.

Má-li být trigr spuštěn pro každý řádek modifikovaný operací UPDATE, má smysl v sekci „Referencing“ v položkách „Old as“ a „New as“ určit, jakými jmény se bude v těle trigru odkazovat na starou a novou hodnotu modifikovaného řádku tabulky. Hodnotou řádku se rozumí hodnoty jednotlivých sloupců či atributů tabulky uložené v daném řádku. Starou hodnotou řádku se zde rozumí původní (modifikovaná) hodnota řádku skutečně uložená v tabulce v databázi. Novou hodnotou řádku se rozumí opravená či modifikovaná hodnota řádku, která je zatím pouze v paměti, ale která se operací UPDATE právě ukládá do tabulky v databázi. Pomocí pojmenování těchto starých a nových hodnot řádku je pak možné zjistit a případně zkontrolovat i staré a nové hodnoty jednotlivých atributů. Podrobněji viz. následující příklad.

A konečně položka „Condition“ může obsahovat podmínku, která omezí spuštění trigru pouze na ty modifikované řádky tabulky, které této podmínce vyhovují.

Příklad databázového trigru

V tomto příkladu předpokládáme dvě tabulky s následující strukturou:

ZBOZI (IDZBOZI: NUMBER, NAZEV: VARCHAR2,)

POHYBY (CISZBOZI: NUMBER, DATUM: DATE, NAKUP: NUMBER, ...)

Tabulka ZBOZI obsahuje informace o jednotlivých druzích zboží a tabulka POHYBY obsahuje informace o jednotlivých nákupech a prodeích konkrétního zboží. Tyto tabulky svázané relací 1:N tedy reprezentují jednoduchý sklad. Je samozřejmě nutné zachovat referenční integritu, tedy zajistit, aby každý pohyb (nákup nebo prodej) patřil k nějakému druhu zboží.

Databázový trigr hlídající toto integritní omezení může vypadat například takto (jsou zde uvedeny jednotlivé položky a jejich hodnoty dle předešlého popisu vytváření databázového trigru, vysvětlení následuje):

- složka „General“
 - položka „Trigger“ - MUJTRIGR
 - položka „Schema“ - XNOVAK01
 - sekce „Trigger On“
 - položka „Schema“ - XNOVAK01
 - položka „Table“ - ZBOZI
 - položka „Trigger Body“

```
UPDATE POHYBY SET CISZBOZI = :NEWVAL.IDZBOZI
WHERE POHYBY.CISZBOZI = :OLDVAL.IDZBOZI;
```
- složka „Timing“
 - sekce „Triggering Statement“
 - přepínací tlačítko „Before“ zapnuto
 - zaškrtačovací box „Insert“ nezaškrtnut
 - zaškrtačovací box „Delete“ nezaškrtnut
 - zaškrtačovací box „Update of Columns“ zaškrtnut, vybrán sloupec IDZBOZI
 - zaškrtačovací box „Trigger for Each Row“ zaškrtnut
 - sekce „Referencing“
 - položka „Old as“ - OLDVAL
 - položka „New as“ - NEWVAL
 - položka „Condition - prázdná

Tento trigr bude fungovat tak, že při každé změně sloupce IDZBOZI v tabulce ZBOZI (tedy při každém přečíslování některého druhu zboží), se přečíslení provede i v tabulce POHYBY ve větách reprezentujících pohyby tohoto přečíslovávaného zboží. Tedy při každém přečíslování MUJTRIGR obě tabulky sesynchronizuje. Jinými slovy, daný druh zboží se přečísluje v obou tabulkách současně a referenční integrita tak zůstane zachována.

V prvních položkách složky „General“ je určeno, že majitelem vytvářeného trigru MUJTRIGR jsem já, a že tento trigr bude hlídat moji tabulku ZBOZI.

Ve složce „Timing“ je zadáno, že trigr se spustí před provedením operace UPDATE, která bude měnit hodnotu sloupce IDZBOZI. Změna názvu zboží nebo jiného údaje v tabulce ZBOZI nevadí, protože tyto další sloupce nemají žádný vliv na tabulku POHYBY. Trigr se spustí pro každý modifikovaný řádek tabulky ZBOZI, tedy zareaguje na každý jednotlivý přečíslovávaný druh zboží, protože musí pro každý takový řádek přečíslovat v tabulce POHYBY odpovídající řádky. Starou a novou hodnotu řádku (v tabulce ZBOZI) si pojmenují OLDVAL a NEWVAL, abych mohl přečíslování v tabulce POHYBY korektně provést (viz. později v popisu těla trigru). Položka „Condition“ zůstane prázdná, protože trigr musí reagovat bezpodmínečně na každý modifikovaný řádek.

V položce „Trigger Body“ je tělo trigru, tedy příkaz UPDATE provádějící synchronizaci tabulek ZBOZI a POHYBY. Klauzule WHERE vybírá řádky z tabulky POHYBY, které budou modifikovány. Musí být vybrány všechny řádky reprezentující pohyby přečíslovávaného zboží, tedy ty řádky z tabulky POHYBY, jejichž číslo zboží (POHYBY.CISZBOZI) je rovno číslu zboží (IDZBOZI), které je přečíslováváno. Ovšem starému, původnímu číslu zboží, které je stále ještě zapsáno v databázi. A v takto vybraných řádcích tabulky POHYBY změním číslo zboží (POHYBY.CISZBOZI) na novou hodnotu čísla zboží (IDZBOZI), která se zapisuje do tabulky ZBOZI. Například pokud se budou přečíslovávat hřebíky z čísla 011 na 311, bude příkaz UPDATE vypadat takto:

```
UPDATE POHYBY SET CISZBOZI = 311 WHERE POHYBY.CISZBOZI = 011;
```

Abych mohl tato dvě čísla, 011 a 311, zjistit, proto si starou a novou hodnotu modifikovaného řádku tabulky ZBOZI pojmenují OLDVAL a NEWVAL. Na číslo zboží se pak neodkazují jako na ZBOZI.IDZBOZI jako obvykle, ale jako na OLDVAL.IDZBOZI a NEWVAL.IDZBOZI podle toho potřebují-li pracovat se

starou (původní a v databázi uloženou) nebo s novou (právě doeditovanou a pouze v paměti se nacházející a do tabulky v databázi právě ukládanou) hodnotou řádku tabulky ZBOZI.

5.5 Uložená procedura

Uložené procedury slouží pro rozdělování výpočetní zátěže mezi klienta a server. Na rozdíl od aplikace, která běží na klientském počítači, jsou totiž uložené procedury vykonávány přímo na serveru.

Postup při vytváření nebo modifikaci uložené procedury

Pokud chceme vytvářet novou uloženou proceduru, klikneme na ikonu „Create“ nebo zvolíme stejnou funkci z menu „Object“ a zvolíme typ vytvářeného objektu „Procedure“. Pokud chceme uloženou proceduru opravovat, najdeme ji v levém okně „Schema Manageru“ a klikneme na její název.

Po zadání všech parametrů nebo po provedení všech změn potvrdíme vytvoření procedury nebo uložení změn stiskem tlačítka „Apply“.

Uložená procedura musí být ještě přeložena, čehož dosáhneme stiskem tlačítka „Recompile“. V sekci „Status“ je pak informace o stavu procedury. Je-li zde zobrazeno hlášení „Valid“ a tlačítko „Show Errors“ je nepřístupné, byla procedura bezchybně přeložena a správně uložena do databáze. Je-li zobrazeno hlášení „Invalid“, pak po tisku tlačítka „Show Errors“ bude zobrazen seznam chyb objevených při překladu procedury. Chyby je nutné opravit, změny uložit a proceduru opět přeložit.

Popis uložené procedury

V zobrazeném dialogovém okně je možné zadat nebo opravit následující údaje.

Ve složce „General“ se zadávají základní údaje o uložené proceduře, zejména v položkách „Name“ (jméno uložené procedury) a „Schema“ (majitel uložené procedury, tedy Vy). V sekci „Source“ se zapisuje definice procedury podobně jako v jazyce PL/SQL. Definice je ve tvaru:

```
( jméno_parametru typ_parametru, ... ) is begin tělo_procedury end ;
```

Vynechává se tedy klíčové slovo PROCEDURE a jméno procedury, jinak je vše stejné jako v jazyce PL/SQL při definici procedury ve formuláři nebo v knihovně. Pokud procedura nemá parametry, kulaté závorky a seznam parametrů se vynechají a začne se klíčovým slovem IS. V těle procedury mohou být použity s jistými omezeními příkazy jazyka SQL i PL/SQL.

Volání uložené procedury

Uloženou proceduru je možné volat z databázového trigru, jiné uložené procedury, z formuláře, menu, zkratka odkudkoliv jako každou jinou proceduru. Volá se svým jménem, případně spolu se jménem majitele, případně se seznamem parametrů:

```
NEJAKAPROCEDURA;  
XNOVAK00.MOJEPROCEDURA;  
XNOVAK00.MOJEPROCEDURA2 ('Josef', 'Novák');
```

6 Forms Designer

Je grafický objektový nástroj pro návrh a tvorbu formulářů, menu a knihoven, tedy jednotlivých komponent koncové aplikace. Nalézá se ve složce „Start“ - „Programs“ - „Developer 2000 for Win95/NT“ - „Forms Designer“. V dalším textu je uveden popis jeho jednotlivých částí.

6.1 Popis Object Navigatoru

V „Object Navigatoru“ jsou zobrazeny všechny důležité komponenty a části „Forms Designeru“.

Forms

Definované nebo editované formuláře.

Menus

Definované nebo editované návrhy systémů menu.

Libraries

Knihovny procedur a funkcí v jazyce PL/SQL. Pokud je nutné ve formuláři nebo menu provádět výpočty, ..., je možné si potřebné rutiny uložit do knihovny a potom je vícenásobně používat.

Build-In Packages

Vestavěné balíky procedur a funkcí, které je možné při programování v jazyce PL/SQL použít.

Database Objects

Přístupné databázové objekty, např. tabulky, pohledy, uložené procedury, apod., rozdělené podle svých vlastníků.

V paletě nástrojů ve sloupci na levém okraji okna „Object Navigatoru“ jsou ikony provádějící jednotlivé funkce editoru.

Open

Otevře existující soubor s definicí formuláře, menu nebo knihovny a zobrazí jeho obsah v okně „Forms Designeru“.

Save

Aktuální editovaný formulář, menu nebo knihovnu uloží do souboru na disku.

Run

Spustí aktuální formulář.

Debug Mode

Zapíná ladicí režim při spuštění formuláře.

Create

Vytvoří novou instanci aktuálního objektu.

Delete

Zruší aktuální instanci objektu.

Expand

Zobrazí všechny potomky aktuálního objektu (vzhledem k hierarchii objektů).

Collapse

Přestane zobrazovat potomky aktuálního objektu.

Expand All

Zobrazí všechny objekty v celé hierarchii objektů.

Collapse All

Zobrazí pouze hlavní objekty a žádné jejich potomky.

6.2 Formulář

Formulář definuje vzhled a funkci výsledné aplikace. Aplikace je tedy tvořena jedním nebo více formuláři.

6.2.1 Popis formuláře

Formulář se skládá z několika různých komponent. Některé z nich musí, jiné mohou být definovány.

Triggers

Trigr je procedura nebo funkce, která se spustí vždy při vzniku nějaké definované události, např. stisku tlačítka. Podle toho, zda je trigr definován u jednotlivého objektu, v bloku nebo formuláři, reaguje pouze na události vzniklé v daném objektu, vzniklé kdekoliv v daném bloku nebo kdekoliv v celém formuláři. Např. reaguje na stisk pouze „svého“ tlačítka, na stisk jakéhokoliv tlačítka v daném bloku nebo na stisk libovolného tlačítka ve formuláři.

Alerts

Upozornění je objekt, který se zobrazí jako okno se specifikovaným textem a tlačítky. Používá se pro zobrazení varování, upozornění nebo zpráv.

Attached Libraries

K formuláři je možné připojit předem definovanou a přeloženou knihovnu a potom v něm používat všechny procedury a funkce v ní definované.

Blocks

Blok je objekt, který na obrazovce výsledné aplikace není vidět, slouží pouze k tomu, aby v sobě obsahoval (sdružoval) objekty logicky patřící k sobě.

Bloky jsou dvou základních typů :

- databázový blok - je svázán s určitou databázovou tabulkou, jejíž položky obsahuje.
- řídicí blok - nemá souvislost se žádnou databázovou tabulkou. Obsahuje položky s databází nesouvisející, např. tlačítka, editační a jiná pole, apod.

Canvas-Views

Plocha viditelná v daném okně. Na ní jsou rozmístěny objekty z jednotlivých bloků.

Windows

Okno na obrazovce aplikace, ve kterém je vidět určitá plocha a na ní rozmístěné objekty.

6.2.2 Popis editoru vzhledu plochy

Editor vzhledu plochy slouží k definici vzhledu plochy zobrazené v některém okně, rozmístění nedefinovaných objektů na tuto plochu.

V paletě nástrojů ve sloupci na levém okraji okna a v řádku nástrojů na horním okraji okna editoru vzhledu plochy jsou ikony provádějící jednotlivé funkce editoru.

Funkce v řádku nástrojů jsou identické s funkcemi v paletě nástrojů „Object Navigator“.

Při používání funkcí z palety nástrojů je vhodné dodržet následující postup :

1. Kliknout levým tlačítkem myši na ikonu zvolené funkce.
2. Kliknout levým tlačítkem myši na místo na pracovní ploše editoru vzhledu plochy, kde má být nový objekt umístěn.
3. Kliknout pravým tlačítkem myši na vytvořený objekt. V zobrazeném okně zvolit položku „Properties“ a nastavit potřebné parametry objektu.

V paletě nástrojů jsou následující funkce :

Select

Zvýrazní zvolený objekt. Se zvýrazněným objektem je možné provádět další operace, např. měnit jeho parametry, přesunovat ho, apod.

Text

Umístí na plochu textový objekt se zadaným textem.

Push Button

Umístí na plochu tlačítko, kterému uživatel může přiřadit specifickou funkci. Děje se tak definicí trigru „When_Button_Pressed“, který se spustí při stisku tlačítka a provede nadefinovanou akci.

Check Box

Umístí na plochu zaškrtačací pole. To může být buď zaškrtnuto nebo nezaškrtnuto, jeho hodnota může být dále využita v aplikaci.

Radio Button

Umístí na plochu přepínací pole. Zároveň je zařadí do skupiny přepínacích polí. V dané skupině může být aktivní pouze jedno přepínací pole. Hodnotu skupiny přepínacích polí (identifikaci aktivního pole) je možné dále využít v aplikaci.

Text Item

Umístí na plochu editační pole. V něm je možné editovat hodnotu zadaného typu převzatou z databázové tabulky nebo přiřazenou programově.

Display Item

Umístí na plochu zobrazovací objekt. Ten pouze zobrazuje zadanou hodnotu definovaného typu.

List Item

Umístí na plochu objekt typu „Seznam hodnot“. Hodnoty je možné do seznamu vybrat z databázové tabulky, přiřadit při vzniku objektu nebo při běhu aplikace. Z těchto hodnot uživatel vybere jednu, která se stane hodnotou celého objektu. Tuto hodnotu objektu je možné v aplikaci dále používat.

Fill Color

Barva výplně objektů.

Line Color

Barva linky. Je vhodné ji nastavit na hodnotu „No Line“, aby na ploše nebyl každý objekt orámován. Zejména orámované texty vypadají nehezky. Nastavení je NUTNÉ provést před definováním objektů.

Text Color

Barva textu umístěného na ploše.

6.2.3 Popis editoru PL-SQL

Editor PL-SQL se používá pro psaní zdrojového textu v jazyce PL-SQL při programování trigrů, položek menu, apod.

V horní části okna editoru je sada tlačítek provádějící jednotlivé funkce editoru :

- Compile - přeloží zdrojový text v jazyce PL-SQL do spustitelného tvaru. Seznam případných chybových hlášení je zobrazen v dolní části okna editoru
- Revert - vrátí zpět poslední provedenou změnu
- New - vytvoří novou instanci aktuálního objektu. Např. při editaci těla trigru vytvoří nový trigr
- Delete - zruší aktuální objekt
- Close - uzavře okno editoru PL-SQL
- Help - zobrazí nápovědu

Pod sadou tlačítek je řada editačních polí typu seznam hodnot, které specifikují aktuální objekt :

- Type - typ editovaného objektu; trigr nebo programová jednotka
- Object - specifikuje aktuální objekt, např. tlačítko, ke kterému trigr patří, a blok, ve kterém je tlačítko definováno
- Name - jméno programové jednotky nebo typ trigru.

Zdrojový text se edituje klasickým způsobem v editačním okně. Přesný popis syntaxe jazyka PL-SQL je uveden v souboru PLSQL.DOC nebo v manuálech či dokumentaci „Forms Designeru“, kde je také seznam vestavěných procedur a funkcí a seznam všech typů trigru.

6.2.4 Důležité parametry objektů formuláře

V této kapitole jsou popsány nejdůležitější parametry objektů. Některé parametry není možné nastavit u některých objektů, u jiných objektů některé parametry nemají smysl nebo vliv na chování objektu.

- Name - jméno objektu, pomocí kterého je objekt identifikován. Tento parametr mají všechny objekty

Alert

- Alert Style - typ upozornění (Stop, Caution, Note). Liší se jen ikonou zobrazenou v okně upozornění
- Button 1,2,3 - Názvy tří tlačítek zobrazených v okně upozornění. Pokud je název prázdný, tlačítko se nezobrazí.
- Default Alert Button - Identifikace implicitního tlačítka
- Message - text zobrazený v okně upozornění
- Title - titulek okna upozornění

Block

Všechny důležité parametry bloku se nastavují při jeho vzniku a není možné je později změnit. Databázový a řídicí blok se liší pouze tím, že řídicí blok nemá zadány parametry vztahující se k databázové tabulce.

Canvas-View

- Width - šířka plochy
- Height - výška plochy
- Window - okno, ve kterém je plocha viditelná
Je třeba dbát na to, aby rozměry plochy odpovídaly rozměrům okna, ve kterém je plocha zobrazena. Jinak může zůstat část plochy přesahující okno skrytá.

Form Module

- Console Window - základní okno aplikace, ve kterém se zobrazí stavový řádek a další informace, a které by mělo zůstat stále zobrazeno
- First Navigation Block - blok, který se stane aktuální po spuštění formuláře. Měl by patřit do základního okna
- Menu Module - jméno souboru obsahujícího spustitelný tvar systému menu patřícího k formuláři

Items

Jsou objekty umístované na jednotlivé plochy, např. tlačítka, zaškrtávací, přepínací, editační a zobrazovací pole, apod. Jejich společné vlastnosti jsou uvedeny v následujícím seznamu :

- Canvas - plocha, na které je objekt zobrazen.
- X Position, Y Position - souřadnice objektu na ploše
- Width, Height - šířka a výška objektu
- Displayed - je-li objekt na ploše viditelný
- Label - název, nadpis nebo popis objektu, který je na nebo u objektu zobrazen. Nezaměňovat se jménem objektu.

Následují další vlastnosti se týkají objektů, které obsahují nějakou hodnotu, jako jsou zaškrťovací, editační, přepínací a zobrazovací pole :

- Data Type - typ datové hodnoty objektu
- Maximum Length - maximální délka hodnoty, např. maximální délka znakového řetězce
- Default Value - implicitní hodnota objektu
- Copy Value from Item - jméno databázové tabulky a její položky, ze které se převezme hodnota objektu. Tímto způsobem je například možné zajistit, že hodnota zobrazená v editačním poli „Jméno zaměstnance“ se převezme z databázové položky JMENO a opravená hodnota se zpět do této položky uloží. Uložení hodnoty do databázové tabulky se provede až po zavolání příkazu COMMIT. Tento způsob funguje pouze pro databázové položky, nikoliv pro proměnné definované ve formuláři. Pro obecnou práci s hodnotou objektu se používají tzv. vázané proměnné (bind variables) - viz. popis PL-SQL nebo demonstrační formulář.

Dále jsou uvedeny parametry specifické pro jednotlivé objekty :

Check Box

- Checked Value - hodnota objektu, pokud je zaškrtnut
- Unchecked Value - hodnota nezaškrtnutého objektu
- Check Box Other Values - jsou-li povoleny jiné hodnoty objektu než hodnoty uvedené v předchozích dvou parametrech

Radio Group

Objekt typu „Skupina přepínacích polí“ (Radio Group) obsahuje několik objektů typu „Přepínací pole“ (Radio Button). Hodnota objektu „Skupina přepínacích polí“ se určí podle hodnoty právě zvoleného objektu „Přepínací pole“.

- Other Values - jsou-li povoleny jiné hodnoty než ty uvedené v jednotlivých objektech skupiny přepínacích polí.

Radio Button

- Value - hodnota objektu typu „Přepínací pole“

List Item

- List Element - zde je možné ručně zadat jednotlivé prvky seznamu hodnot. Prvky tohoto seznamu je také možné vybrat pomocí SQL příkazu SELECT
- List Style - určuje typ seznamu. Typy „PopList“ a „Combo Box“ zabírají na obrazovce méně místa, ale je vidět jen jedna hodnota ze seznamu. Teprve při jeho otevření se zobrazí více hodnot. Typ „Tlist“ zobrazuje současně více hodnot seznamu

Relation

Tato skupina parametrů se týká tzv. Master/Detail bloků (hlavní a podřízený blok), tedy bloků tabulek spojených relací.

- Detail Block - blok patřící sekundární databázové tabulce. Např. hlavní blok patří tabulce položek objednávky obsahující také číslo objednaného zboží a podřízený blok patří tabulce číselníku zboží. Obě tabulky jsou spojeny relací pomocí čísla zboží.
- Join Condition - podmínka realizující relaci. Např. rovnost čísla zboží v číselníku a objednávce.
- Coordination - způsob vyhledávání a zobrazování vět v sekundární tabulce. Nastavuje se pomocí dvou parametrů : „Deferred“ a „Auto Query“.
 - Pokud je parametr „Deferred“ nastaven na „False“, data v podřízeném bloku jsou vyhledána a zobrazena ihned poté, co se změní aktuální věta v hlavním bloku. Např. při přechodu na jiné zboží v objednávce se ihned v sekundární tabulce vyhledá a zobrazí jeho název. Tento implicitní způsob je vhodný, pokud jsou oba bloky současně zobrazeny.

- Pokud jsou oba tyto parametry nastaveny na hodnotu „True“, jsou data z podřízené tabulky vyhledána a zobrazena až tehdy, když se podřízený blok stane aktuální. Např. až se otevře okno, ve kterém je podřízený blok zobrazen.
- Když je parametr „Deferred“ nastaven na „True“ a parametr „Auto Query“ na „False“, data jsou vyhledána a zobrazena až poté, co se stane podřízený blok aktuální a uživatel o vyhledání explicitně požádá.
- Master Deletes - specifikuje jak ovlivní zrušení věty v hlavním bloku věty v podřízeném bloku.
 - Non-Isolated - implicitně nastaví, které zakazuje zrušení věty v hlavním bloku, pokud jsou s ní relací spojeny nějaké věty v podřízeném bloku.
 - Isolated - zrušení věty v hlavním bloku nijak neovlivní věty v podřízeném bloku.
 - Cascading - při zrušení věty v hlavním bloku se automaticky zruší věty v podřízeném bloku, které jsou s ní svázány relací.
- Prevent Masterless Operations - zakazuje vyhledávání a vkládání vět v podřízeném bloku, pokud není v hlavním bloku zvolena odpovídající věta.

Window

- X Position, Y Position - určují souřadnice levého horního rohu okna na obrazovce
- Width, Height - šířka a výška okna
- Title - titulek (nadpis) okna
- Window Style - okno typu „Document“ je normální okno. Okno typu „Dialog“ nemůže být překryto ostatními okny. Spolu s parametrem „Modal“ je vhodné pro okna, kde se zadávají důležité informace, např. dialogová okna, apod.
- Modal - Pokud je nastaven tento parametr, okno je možné zavřít nebo opustit jen po zásahu uživatele. Tento parametr je vhodné použít spolu s nastavením „Window Style“ na „Dialog“.

6.2.5 Postup při tvorbě formuláře

Připojení k databázi

Spustíme funkci „File“ -> „Connect“, zadáme své jméno, heslo a databázi, do které se připojíme. Jméno databáze je vždy „UIVTstud“, a to včetně velkých a malých písmen!

Vytvoření formuláře

Pokud systém Oracle prázdný formulář nevytvořil implicitně, vytvoříme jej pomocí funkce „Create“ v „Object Navigatoru“. Nebo pomocí funkce „Open“ načteme existující formulář ze souboru na disku.

Vytvoření databázového bloku

Nejdříve je nutné vytvořit databázový blok, který bude obsahovat položky ze zvolené databázové tabulky, aby formulář mohl s touto tabulkou pracovat.

Při vytváření databázového bloku je nutné dodržet následující postup:

1. V „Object Navigatoru“ zvolíme funkci „Create“. Objeví se okno, ve kterém se zadávají parametry vytvářeného bloku.
2. Ve složce „General“ je nutné nejprve zadat databázovou tabulku, ke které bude blok patřit. To provedeme stiskem tlačítka „Select“ vedle editačního pole „Base Table“. V okně „Tables“, které se objeví zadáme, kde a jak se bude tabulka hledat. Zda chceme nebo nechceme vyhledat tabulky i jiných vlastníků, zda chceme hledat pouze tabulky nebo i pohledy. Po stisku tlačítka „Ok“ si ze zobrazeného seznamu vybereme požadovanou tabulku a potvrdíme výběr stiskem tlačítka „Ok“. Jméno zvolené tabulky se objeví v poli „Base Table“ v okně parametrů bloku.
3. Zadáme jméno bloku.
4. Zkontrolujeme, zda blok vzniká a bude zobrazen na správné ploše.

5. Přejdeme do složky „Items“. Zde se vybírají databázové položky ze zvolené tabulky, které budou do bloku patřit. Po stisku tlačítka „Select Columns“ se provede výběr požadovaných položek.
6. Ve složce „Layout“ se určí, jak bude blok vypadat. V editačním poli „Records“ je možné zadat, kolik databázových vět současně bude v bloku zobrazeno a v poli „Spacing“ jak budou jednotlivé položky od sebe vzdáleny. Po zaškrtnutí pole „Button Palette“ ve sloupci „Options“ systém Oracle doplní k bloku sadu implicitních tlačítek, která usnadní manipulaci s databázovými větami ve výsledné aplikaci. Po zaškrtnutí pole „Scroll Bar“ systém Oracle k zobrazeným databázovým větám přiřadí posuvný řádek (scroll bar).
7. Složka „Master/Detail“ slouží pro vytváření bloků pro tabulky svázané do relace.
8. Teprve po zadání a zkontrolování všech těchto parametrů je možné stisknout tlačítka „Ok“ a blok skutečně vytvořit.

Upozornění :

Parametry zadávané při vytváření bloku není možné později měnit. Proto je důležité je pečlivě zadat a zkontrolovat. Při pozdějším odhalení chyby nebo nutnosti změny je nezbytné blok zrušit a vytvořit znovu.

Poznámka :

Uvedený postup platí pro vytváření databázového bloku. Při vytváření řídicího bloku se žádný parametr týkající se databázové tabulky nenastavuje a zůstává prázdný.

Vytváření dalších potřebných objektů

Je-li třeba, je možné vytvořit pomocí funkce „Create“ v „Object Navigatoru“ nebo přímo v editoru vzhledu plochy další objekty. Po kliknutí pravým tlačítkem myši na nový objekt a volbě funkce „Properties“ z objeveného se menu je nutné nastavit parametry objektu na požadované hodnoty.

Upozornění :

Při vytváření objektu je vždy nutné dbát na to, aby vznikl ve správném bloku a byl zobrazen na správné ploše ve správném okně.

Tvorba vzhledu jednotlivých definovaných ploch

V „Layout Editoru“ (editoru vzhledu plochy) se provádí umístění jednotlivých objektů na plochu, případně se vytváří a na plochu umísťují objekty nové.

Upozornění :

Před začátkem editace vzhledu plochy je velice vhodné zrušit barvu linky (nastavit v „Layout Editoru“ barvu linky (Line Color) na „No Line“). Jinak budou objekty v „Layout Editoru“ orámovány.

Uložení definice formuláře do souboru

Funkcí „File“ -> „Save“ nebo „Save As“ provedeme uložení formuláře do souboru na disku. Standardní přípona je .fmb.

Upozornění :

Před spuštěním formuláře je NUTNÉ formulář nejprve uložit do souboru na disku. Oracle jinak formulář před spuštěním uloží sám do svého implicitního adresáře, což pod Windows NT způsobí chybu díky zákazu zápisu do tohoto adresáře.

Vygenerování spustitelného tvaru formuláře

Funkce „File“ -> „Administration“ -> „Generate“ vygeneruje spustitelný tvar formuláře a uloží ho do souboru s příponou .fmx.

Druhou možností je formulář přímo spustit funkcí „File“ -> „Run“.

6.3 Menu

6.3.1 Tvorba menu

Menu se vytvářejí standardním postupem. V „Object Navigatoru“ se pomocí funkce „Create“ vytvoří nový modul menu. V okně „Properties“ je nutné nastavit jeho jméno a, pokud tak neučinil systém Oracle sám, jméno hlavního menu v tomto modulu.

V modulu menu je možné vytvářet vlastní programové jednotky (Program Units). V nich je možné ukládat funkce a procedury opakovaně používané v modulu menu.

Druhá a lepší varianta je vytvořit si vlastní knihovnu procedur a funkcí a v ní si potřebné procedury a funkce nadefinovat. Způsob vytváření knihoven je popsán v samostatné kapitole. Potřebnou knihovnu je možné pomocí objektu „Attached Libraries“ modulu menu k tomuto modulu připojit a v nich nadefinované procedury a funkce používat.

6.3.2 Popis menu editoru

V menu editoru se provádí definice a editace systému menu.

V paletě nástrojů v řádku na horním okraji okna menu editoru jsou ikony provádějící jednotlivé funkce editoru.

Create Down

Vytvoří novou položku menu pod aktuální položkou.

Create Right

Vytvoří novou položku menu vpravo od aktuální položky.

Delete

Zruší aktuální položku menu.

Pomocí těchto funkcí je možné vytvořit libovolně složitý systém menu. Každá položka menu může mít svoje individuálně nastavené parametry (Properties).

Důležité parametry položky menu :

- Name - jméno objektu „Položka menu“
- Menu Item Type - typ položky menu
 - Plain - normální položka menu
 - Check - logická položka menu. Má hodnotu TRUE nebo FALSE (zaškrtnuta nebo nezaškrtnuta). Při její volbě pouze mění svoji hodnotu
 - Radio - logická položka menu, která je částí skupiny přepínacích položek (Radio Group). Při její volbě se její hodnota nastaví na TRUE a hodnota ostatních položek ve skupině na FALSE. Zároveň se může spustit akce, pokud ne definovaná.
 - Separator - slouží k optickému oddělení položek v menu. Zobrazí se jako horizontální čára dělicí menu.
 - Magic - položka menu speciálního typu. Je možné jí přiřadit jednu z předdefinovaných standardních funkcí (Cut, Copy, Paste, Clear, Undo, About, Help, Quit nebo Window).
- Command Type - typ příkazu (akce), který tato položka provádí
 - Null - neprovádí žádnou akci (povinné pro položku typu Separator)
 - Menu - zobrazí submenu
 - PL-SQL - provede akci naprogramovanou v jazyce PL-SQL
- Command Text - zdrojový text v jazyce PL-SQL, který specifikuje akci prováděnou položkou menu
- Magic Item - pokud je položka menu typu „Magic“, zde se specifikuje jakou předdefinovanou standardní akci bude provádět

- Menu Item Radio Group - pokud je položka menu jméno typu „Radio“, zadává se zde jméno skupiny přepínacích položek, do které tato položka menu patří
- Label - zobrazovaný název položky menu
- Hint - text zobrazovaný ve stavovém řádku jako zpráva (Message), když je tato položka menu aktuální
- Help - text zobrazený po vyvolání vestavěné procedury HELP, když je tato položka menu aktuální

6.3.3 Postup při návrhu menu

Připojení k databázi

Pokud již nejsme připojeni k databázi, spustíme funkci „File“ -> „Connect“, zadáme své jméno, heslo a databázi, do které se připojujeme. Jméno databáze je vždy „UIVTstud“, a to včetně velkých a malých písmen!

Tvorba systému menu

V „Object Navigatoru“ pomocí funkce „Create“ vytvoříme nový modul menu nebo pomocí funkce „Open“ otevřeme existující modul menu. Spustíme menu editor a v něm vytváříme nebo editujeme systém menu.

Uložení definice menu do souboru

Funkcí „File“ -> „Save“ nebo „Save As“ provedeme uložení definice menu do souboru na disku. Standardní přípona je .mmb.

Vygenerování spustitelného tvaru menu

Funkce „File“ -> „Administration“ -> „Generate“ vygeneruje spustitelný tvar systému menu a uloží ho do souboru s příponou .mmx.

Připojení vytvořeného menu k formuláři

Dokud není vytvořený a přeložený systém menu připojen k formuláři, formulář (aplikace) o něm neví a dál běží s implicitním menu Oraclu.

Menu se k formuláři připojí tak, že se zadá úplné jméno souboru, který obsahuje přeložený tvar menu, (včetně disku a cesty) do parametru „Menu Module“ objektu typu Form, který reprezentuje daný formulář.

6.4 Knihovny

Knihovna slouží k uchovávání programových jednotek. Obsahuje jednotlivé programové jednotky, např. procedury, funkce, balíky procedur a funkcí. Uložení logicky souvisejících programových jednotek do jedné knihovny lze dosáhnout lepší a přehlednější organizace výsledné aplikace.

6.4.1 Tvorba knihoven

Knihovny se vytvářejí standardním postupem. V „Object Navigatoru“ se pomocí funkce „Create“ vytvoří nová knihovna. Při ukládání nové knihovny pomocí funkce „File“ -> „Save As“ se určí jméno souboru, ve kterém bude knihovna uložena, což je zároveň také jménem knihovny.

V knihovně je možné vytvářet vlastní programové jednotky (Program Units). V nich je možné ukládat funkce a procedury opakovaně používané.

Do knihovny je také možné připojit další knihovny a ve vytvářených programových jednotkách požívat funkce a procedury z připojených knihoven.

6.4.2 Postup při tvorbě knihoven

Připojení k databázi

Pokud již nejsme připojeni k databázi, spustíme funkci „File“ -> „Connect“, zadáme své jméno, heslo a databázi, do které se připojujeme. Jméno databáze je vždy „UIVTstud“, a to včetně velkých a malých písmen!

Tvorba knihovny

V „Object Navigatoru“ pomocí funkce „Create“ vytvoříme novou knihovnu nebo pomocí funkce „Open“ otevřeme existující knihovnu. Pomocí editoru PL-SQL programujeme nebo opravujeme programové jednotky knihovny.

každou nově vytvořenou nebo opravenou programovou jednotku je nutné přeložit.

Uložení knihovny do souboru

Funkcí „File“ -> „Save“ nebo „Save As“ provedeme uložení knihovny do souboru na disku. Standardní přípona je .pll.

Vygenerování spustitelného tvaru knihovny

Funkce „File“ -> „Administration“ -> „Generate“ vygeneruje spustitelný tvar knihovny a uloží ho do souboru s příponou .plx.

7 Report Designer

Je grafický objektový nástroj pro návrh a tvorbu výstupních sestav. Nalézá se ve složce „Start“ - „Programs“ - „Developer 2000 for Win 95/NT“ - „Report Designer“. V dalším textu je uveden popis jeho jednotlivých částí.

7.1 Popis Object Navigatoru

V „Object Navigatoru“ jsou zobrazeny všechny důležité komponenty a části „Report Designeru“.

Reports

Definované nebo editované definice výstupních sestav.

External Queries

Externí dotazy uložené odděleně; vhodné pro vícenásobně používané dotazy.

Libraries

Knihovny procedur a funkcí v jazyce PL-SQL. Pokud je nutné v sestavě provádět výpočty, ..., je možné si potřebné rutiny uložit do knihovny a potom je vícenásobně používat.

Build-In Packages

Vestavěné balíky procedur a funkcí, které je možné při programování v jazyce PL-SQL použít.

Database Objects

Přístupné databázové objekty, např. tabulky, pohledy, uložené procedury, apod., rozdělené podle svých vlastníků.

V paletě nástrojů ve sloupci na levém okraji okna „Object Navigatoru“ jsou ikony provádějící jednotlivé funkce editoru.

Open

Otevře existující soubor s definicí sestavy a zobrazí jeho obsah v okně „Report Designeru“.

Save

Aktuální editovanou definici sestavy uloží do souboru na disku.

Run

Spustí aktuální definici sestavy a vytiskne sestavu (na obrazovku).

Create

Vytvoří novou instanci aktuálního objektu.

Delete

Zruší aktuální instanci objektu.

Expand

Zobrazí všechny potomky aktuálního objektu (vzhledem k hierarchii objektů).

Collapse

Přestane zobrazovat potomky aktuálního objektu.

Expand All

Zobrazí všechny objekty v celé hierarchii objektů.

Collapse All

Zobrazí pouze hlavní objekty a žádné jejich potomky.

7.2 Popis editoru datového modelu

Editor datového modelu slouží k definici všech datových objektů, které budou v sestavě k dispozici.

V paletě nástrojů ve sloupci na levém okraji okna editoru datového modelu jsou ikony provádějící jednotlivé funkce editoru.

Při používání těchto funkcí je vhodné dodržet tento postup :

1. Kliknout levým tlačítkem myši na ikonu zvolené funkce.
2. Kliknout levým tlačítkem myši na místo na pracovní ploše editoru datového modelu, kde má být nový objekt umístěn. Umístění objektu na pracovní ploše editoru datového modelu nemá nic společného s umístěním objektu v sestavě. Cílem je datové objekty na pracovní ploše editoru datového modelu přehledně umístit s respektováním jejich případné hierarchie, aby byl datový model sestavy jasný a přehledný.
3. Kliknout pravým tlačítkem myši na vytvořený objekt. V zobrazeném okně zvolit položku „Properties“ a nastavit potřebné parametry objektu.

Query

Zadávat dotaz na databázi, který vybírá data z databázové tabulky. Takto vybraná data se budou tisknout ve výstupní sestavě.

Důležité parametry se zadávají ve složce „General“ :

- Name - jméno objektu „Dotaz“
- Maximum Rows - maximální počet řádků, které dotaz může vybrat. Nezádá-li se žádná hodnota, není tento počet omezen.
- External Query - zde je možné zadat jméno dříve vytvořeného externího dotazu.
- SELECT Statement - zde se zadává SQL příkaz SELECT, který vybírá data tištěná ve výstupní sestavě z databázové tabulky

Summary Column

Vytvoří objekt obsahující hodnotu spočtenou z dat v databázové tabulce.

Důležité parametry se zadávají ve složce „General“ :

- Name - jméno objektu
- Function - funkce, která se použije pro výpočet hodnoty objektu. Je možné použít některou ze standardních funkcí, např. součet, počet nebo průměr hodnot, apod.
- Data-Selection - zde se uvádí parametry výpočtu :
 - Source - jméno položky databázové tabulky, nad jejímiž hodnotami bude proveden výpočet
 - DataType, Width - typ a velikost databázové položky, resp. spočtené hodnoty

- Value if Null - hodnota, která se má zobrazit, pokud je hodnota výsledku výpočtu nulová
- Reset At - hodnota „Report“ znamená, že se výpočet provede pro celou sestavu, hodnota „Page“ znamená, že se výpočet provede zvlášť pro každou stránku sestavy
- Break Order - položka se bude tisknout pouze, pokud se změní její hodnota

Data Link

Slouží ke spojení dvou nebo více databázových tabulek do relace.

Formula Column

Vytvoří objekt obsahující hodnotu spočtenou libovolným způsobem. Tento objekt je podobný objektu „Summary Column“ pouze s tím rozdílem, že funkci počítající jeho hodnotu je nutné naprogramovat v jazyce PL-SQL.

7.3 Popis editoru vzhledu sestavy

Editor vzhledu sestavy slouží k definici vzhledu sestavy, rozmístění nadefinovaných objektů do sestavy.

V paletě nástrojů ve sloupci na levém okraji okna a v řádku nástrojů na horním okraji okna editoru vzhledu sestavy jsou ikony provádějící jednotlivé funkce editoru.

V řádku nástrojů jsou následující funkce :

Default Layout

Vytvoří implicitní sestavu, rozmístí objekty v sestavě standardním způsobem. pokud nějaký vzhled sestavy existuje, je PŘEPSÁN.

Body

Zobrazí v pracovní ploše editoru vzhled sestavy.

Margin

Zobrazí v pracovní ploše editoru vzhled sestavy i s okraji sestavy.

Header

Zobrazí v pracovní ploše editoru vzhled záhlaví sestavy. Záhlaví sestavy se tiskne pouze jednou za začátku tisku sestavy.

Trailer

Zobrazí v pracovní ploše editoru vzhled patičky sestavy. Patička sestavy se tiskne pouze jednou za konci tisku sestavy.

Při používání funkcí z palety nástrojů je vhodné dodržet následující postup :

1. Kliknout levým tlačítkem myši na ikonu zvolené funkce.
2. Kliknout levým tlačítkem myši na místo na pracovní ploše editoru vzhledu sestavy, kde má být nový objekt umístěn.
3. Kliknout pravým tlačítkem myši na vytvořený objekt. V zobrazeném okně zvolit položku „Properties“ a nastavit potřebné parametry objektu.

V paletě nástrojů jsou následující funkce :

Select

Zvýrazní zvolený objekt. Se zvýrazněným objektem je možné provádět další operace, např. měnit jeho parametry, přesunovat ho, apod.

Frame Select

Zvýrazní celý rámec i se všemi jeho objekty.

Text

Umístí do sestavy textový objekt se zadaným textem.

Frame

Umístí do sestavy nový rámeček. Ten v sobě může obsahovat další objekty a sdružuje je tak do skupiny. Parametry rámce tak platí i pro všechny jeho objekty. Např. pokud se nastaví parametry rámce tak že se tiskne na začátku každé stránky sestavy, tisknou se na začátku stránky i jeho objekty. Tak je možné tisknout např. záhlaví stránky.

Repeating Frame

Tiskne svůj obsah opakovaně pro každou databázovou větu ze zadané skupiny. Skupina vět vznikne při zadání dotazu v editoru datového modelu.

Field

Umístí do sestavy objekt obsahující hodnotu zvoleného objektu z datového modelu sestavy, např. položku databázové tabulky.

Button

Umístí do sestavy tlačítko, kterému uživatel může přiřadit specifickou funkci.

Anchor

Nastaví explicitní vazbu otec - syn mezi dvěma objekty v sestavě. Nejprve se jednou klikne levým tlačítkem myši na hranu synovského objektu, potom se dvakrát klikne levým tlačítkem myši na hranu otcovského objektu. Vazby otec - syn slouží k definici vzájemných vztahů mezi objekty a určování způsobu tisku jednotlivých objektů. Podrobnější popis je uveden v kapitole „Popis parametrů objektu“.

Fill Color

Barva výplně objektů. Je vhodné ji nastavit na hodnotu „No Fill“, aby se překrývající se objekty nezakrývaly. Nastavení je NUTNÉ provést před definováním objektů.

Line Color

Barva linky. Je vhodné ji nastavit na hodnotu „No Line“, aby ve vytištěné sestavě nebyl každý objekt orámován. Sestava s orámovanými objekty je značně nepřehledná. Nastavení je NUTNÉ provést před definováním objektů.

Text Color

Barva textu v sestavě.

7.4 Popis parametrů objektu

V této kapitole jsou popsány nejdůležitější parametry objektů. Některé parametry není možné nastavit u některých objektů, u jiných objektů některé parametry nemají smysl nebo vliv na chování objektu.

Object

Tato složka se zobrazuje pouze u objektů souvisejících s databázovou tabulkou, tedy u objektů typu „Field“ a „Repeating Frame“.

U objektu typu „Field“ obsahuje tyto parametry :

- Source - zde se zadává, odkud je převzata hodnota objektu.
 - From - zadává se jméno datového objektu definovaného v datovém modelu sestavy.
- Display - zadává se způsob zobrazení hodnoty objektu.
 - Display Mask - maska zobrazení hodnoty objektu.
 - Minimum Window Lines - minimální počet řádků obsahujících hodnotu objektu, které musí být na téže stránce.

U objektu typu „Repeating Frame“ obsahuje tyto parametry :

- Source - jméno skupiny, jejíž databázové položky budou v tomto rámci tištěny.
- Print Direction - směr tisku jednotlivých instancí opakovacího rámce.
- Maximum Records per Page - maximální počet instancí opakovacího rámce na jedné stránce sestavy.
- Minimum Window Records - minimální počet instancí opakovacího rámce na jedné stránce sestavy.
- Spacing - velikost mezer mezi jednotlivými instancemi opakovacího rámce.

General Layout

Tato složka se zobrazuje u všech objektů a udává obecný vzhled objektu a jeho umístění v sestavě.

- Pagination - zde se zadávají podmínky specifikující umístění objektu na stránce sestavy.
 - Page Break Before - před tiskem objektu se vždy provede přechod na novou stránku, objekt tedy bude vždy na začátku nové stránky.
 - Page Break After - po tisku objektu se provede přechod na novou stránku.
 - Page Protect - objekt a celý jeho obsah musí být na jedné stránce. Není-li na aktuální stránce dostatek místa pro tisk objektu, provede se přechod na novou stránku.
 - Keep with Anchoring Object - objekt a jeho otcovský objekt musí být na téže stránce.
- Format Trigger - je funkce určující zda se objekt bude tisknout.
- Print Condition - zde se zadává vztah objektu k jiným objektům. Určuje se zda, kdy a jak se objekt bude tisknout, ale v závislosti na tisku jiného objektu.
 - Object - zde se zadává tzv. referenční objekt, tedy objekt, podle kterého se bude řídit tisk tohoto (aktuálního) objektu.
 - Enclosing - referenčním (řídícím) objektem je objekt obklopující aktuální objekt, např. řídícím objektem objektu typu „Field“ bude rámec, který jej obklopuje.
 - Anchoring - referenčním objektem je otcovský objekt vzhledem k hierarchii objektů v sestavě.
 - Type - zde se určuje kdy se objekt bude tisknout, ovšem vzhledem k referenčnímu objektu. Objekt se bude tisknout :
 - All - při každém tisku referenčního objektu
 - All but First - při každém kromě prvního tisku referenčního objektu
 - First - pouze při prvním tisku referenčního objektu
 - Default - systém Oracle podle svých pravidel sám určí, kdy se objekt bude tisknout
- Sizing - zde se určují rozměry objektu, vertikální a horizontální.
 - Fixed - rozměry objektu jsou pevné a nemění se
 - Contract - vyžadují-li tištěná data objektu menší prostor, než je jeho velikost, velikost objektu se zmenší. V opačném případě se vytiskne tolik dat, kolik se do rozměru objektu vejde.
 - Expand - opačně než Contract. Velikost objektu se může zvětšit, ale ne zmenšit.
 - Variable - rozměry objektu se při tisku sestavy mění podle aktuální velikost tištěných dat.

Upozornění :

Hierarchie objektů se definuje dvěma způsoby :

1. Explicitní určení vztahu otec - syn mezi objekty.
2. Implicitní vazby typu otec - syn mezi objekty definované podle pravidel pro tvorbu implicitních vazeb. Podrobný popis těchto pravidel je uveden v manuálu nebo dokumentaci „Report Designeru“.

7.5 Postup při návrhu sestavy

Připojení k databázi

Spustíme funkci „File“ -> „Connect“, zadáme své jméno, heslo a databázi, do které se připojujeme. Jméno databáze je vždy „UIVTstud“, a to včetně velkých a malých písmen!

Tvorba datového modelu

Spustíme editor datového modelu. Nejprve vytvoříme dotazy vybírající data z tabulky. Potom můžeme vytvořit další datové objekty, jako například součty, počty, počítané položky, apod.

Tvorba vzhledu sestavy

V „Layout Editoru“ (editoru vzhledu sestavy) se provádí umístění jednotlivých objektů do sestavy.

Pomocí funkce „Tool“ -> „Default Layout“ vznikne implicitní sestava s objekty rozmístěnými standardním způsobem. Konkrétní vzhled implicitní sestavy je možné určit pomocí parametrů.

Vzhled implicitní sestavy je možné dále editovat a upravovat.

Upozornění :

Před začátkem editace vzhledu sestavy je velice vhodné zrušit barvu výplně a linky (nastavit v „Layout Editoru“ barvu výplně (Fill Color) na „No Fill“ a barvu linky (Line Color) na „No Line“). Jinak se budou objekty v „Layout Editoru“ vzájemně zakrývat a ve vytištěné sestavě budou objekty orámovány.

Uložení definice sestavy do souboru

Funkcí „File“ -> „Save“ nebo „Save As“ provedeme uložení definice sestavy do souboru na disku. Standardní přípona je .rdf.

Upozornění :

Před spuštěním tisku sestavy je NUTNÉ definici sestavy nejprve uložit do souboru na disku. Oracle jinak definici sestavy před spuštěním uloží sám do svého implicitního adresáře, což pod Windows NT způsobí chybu díky zákazu zápisu do tohoto adresáře.

Vygenerování spustitelného tvaru sestavy

Funkce „File“ -> „Administration“ -> „Generate“ vygeneruje spustitelný tvar definice sestavy a uloží ho do souboru s příponou .rep.

Druhou možností je spustit tisk sestavy ihned funkcí „File“ -> „Run“, zejména pro účely ověření, že vypadá tak, jak jsme chtěli.

8 Data Manager

Program „Data Manager“ slouží pro export a import dat z Oraclovské databáze do souboru a zpět. Tato operace je vhodná zejména pro archivování dat nebo jejich přenos ze školy domů a zpět. Nachází se ve složce „Start“ - „Programs“ - „Oracle Enterprise Manager“ - „Data Manager“.

Po spuštění je nutné se připojit do databáze. Zadáme své jméno, heslo a databázi, do které se připojujeme. Jméno databáze je vždy „stud“.

Ve složce „Export“ nebo „Import“ zadáte soubor kam se budou data exportovat nebo odkud importovat.

POZOR! Soubor je nutné vybrat pomocí tlačítka „Browse“. Pokud zadáte cele jméno souboru přímo do editačního pole, export zhavaruje.

Dále v sekci „To Be Exported“, resp. „To Be Imported“ zvolíte, kterých objektů se bude operace týkat. Zaškrtnete-li volbu „Tables“, bude se týkat pouze tabulek. Stiskem tlačítka „Specify“ pak musíte určit, kterých tabulek (nejčastěji Vašich). Zaškrtnete-li volbu „Users“, bude se týkat všech databázových objektů, tedy nejen tabulek, ale i indexů, generátorů sekvencí, trigrů, procedur, apod. Stiskem tlačítka „Specify“ pak musíte

určit vlastníka těchto objektů (nejčastěji Vy sami). Konečně stiskem tlačítka „Export“, resp. „Import“ se spustí požadovaná operace.

9 Popis demonstračního příkladu

9.1 Popis jednotlivých souborů :

- ex_form.fmb - binární (editovatelný) soubor obsahující demonstrační formulář
- ex_form.fmx - spustitelný (needitovatelný) soubor obsahující demonstrační formulář
- ex_menu.mmb - binární (editovatelný) soubor obsahující menu k demonstračnímu formuláři
- ex_menu.mmx - spustitelný tvar systému menu k demonstračnímu formuláři
- ex_rep.rdf - binární (editovatelný) soubor obsahující demonstrační výstupní sestavu k demonstračnímu formuláři
- ex_rep.rep - spustitelný tvar demonstrační výstupní sestavy k demonstračnímu formuláři

9.2 Doporučení :

Soubory si zkopírujte na svůj disk P: do jeho kořenového adresáře a tam si vyzkoušejte co a jak demonstrační aplikace dělá. Pokud budete mít souboru umístěny v jiném adresáři, je nutné ve všech objektech, kde je uvedena cesta „p:\...” jí zaměnit za správnou specifikaci cesty k souborům.

Jedná se o nastavení cesty k souboru obsahujícímu spustitelný tvar menu. Cesta se nastavuje v parametru „Menu Module“ ve vlastnostech objektu „Ex_Form“. Po kliknutí pravým tlačítkem myši na tento objekt a zvolení funkce „Properties“ ze zobrazeného menu se objeví okno vlastností objektu, kde je možné zmíněný parametr nastavit. Druhým místem, kde je nutné opravit cestu, je tělo trigru „When_Button_Pressed“ objektu (tlačítka) „Report“ v bloku „Button_Palette“. V těle tohoto trigru se volá procedura Run_Product, která má jako jeden ze svých parametrů jméno souboru obsahujícího spustitelný tvar definice sestavy.

Poznámka :

Při práci s demonstračním formulářem není možné pracovat s databázovou tabulkou NOVACEKV.PRAC. Při přímém zadání SQL dotazu v programu „SQL*Plus“ jsou data z této tabulky přístupná, ale přes formulář ne. Příčina této chyby je tedy někde v systému Oracle a pracuje se na jejím odstranění.

9.3 Komentáře

Všechny objekty mají v položce „Comment“ uložen komentář jejich funkce a důležitých parametrů. Komentář je možné si zobrazit po nastavení myši na objekt, klepnutím na pravé tlačítko a výběru funkce „Properties“ ze zobrazeného menu. V okénku „Properties“ je vždy složka či položka „Comment“.

Komentáře podrobněji popisují vlastnosti, parametry a funkci konkrétních objektů. Další popisy konkrétních parametrů a vlastností jsou buď v tomto textu nebo v manuálech či dokumentaci příslušného programového nástroje.

9.4 Funkce demonstrační aplikace

Demonstrační aplikace uložená ve výše uvedených souborech nemá žádnou praktickou funkci. Neprovádí ani žádnou rozumnou práci. Obsahuje pouze fragmenty kódu, které demonstrují práci s konkrétními objekty nebo používání konkrétních funkcí, procedur a vlastností a parametrů objektů. Slouží pouze k předvedení používání některých objektů, nástrojů, tvorby a vzhledu aplikace. Tuto demonstrační aplikaci není možné prakticky používat,

ale postupy použité v této aplikaci je možné použít při tvorbě vlastní aplikace.

Popis demonstračního formuláře

Popis je rozdělen do dvou částí. První popisuje jednotlivé komponenty formuláře a druhá jeho funkci.

Přímo ve formuláři, nikoliv v nějakém bloku nebo tlačítku, jsou definovány dva trigry : Pre-Form a Post-Form. Provádí definované akce před a po spuštění formuláře. Tyto akce jsou okomentovány přímo v jejich tělech.

Upozornění ALERT je definováno pouze pro demonstrování jeho použití. Zobrazí se po stisku tlačítka „Alert“ v základním okně a vrácená hodnota je zobrazena v kontrolním okně.

Dále jsou ve formuláři definována tři okna a tři plochy v jednotlivých oknech viditelné. Okno Window0 je základním oknem formuláře. Jsou v něm na ploše Canvas2 zobrazeny věty z databázové tabulky novacekv.prac, sada tlačítek a editační pole. Okno DialogWin je dialogové okno, ve kterém jsou na ploše DialogCnvs zobrazeny zaškrtačovací a přepínací pole. Okno ChkWin slouží k zobrazení kontrolních informací. Na ploše Canvas_Chk jsou viditelné zobrazovací objekty.

Posledními komponentami jsou bloky. Blok Prac je tzv. „Base Table Block“, tedy databázový blok. Obsahuje databázové položky z tabulky novacekv.prac. Je umístěn na ploše Canvas2 v základním okně. Ostatní bloky jsou tzv. řídicí bloky. Nemají žádnou souvislost s databázovou tabulkou, obsahují objekty pro zobrazování či zadávání informací, výpočty, apod. Blok Button_Palette obsahuje sadu tlačítek pro manipulaci s databázovými větami, práci s okny, tisk sestavy a zobrazení upozornění a editační pole. Je také zobrazen na ploše Canvas2 v základním okně. V bloku DialogBlk jsou definovány dva trigry Pre-Block a Post-Block, které provádějí zadané akce před vstupem do bloku a po jeho opuštění. Dále jsou v tomto bloku zaškrtačovací a přepínací pole a sada tlačítek. Je zobrazen na ploše DialogCnvs v dialogovém okně. Poslední je blok ChkBlock. V něm je nedefinována řada zobrazovacích objektů, které zobrazují zadané hodnoty a tlačítko, které slouží pro zavření kontrolního okna, ve kterém je tento blok zobrazen.

Po spuštění demonstračního formuláře je zobrazeno základní okno. V něm jsou zobrazena editační pole pro zadávání hodnot databázových položek tabulky novacekv.prac. Pod nimi je sada tlačítek a editační pole pro zadání textu.

Po spuštění je formulář v tzv. editačním režimu. V tomto režimu je možné zadávat nové, editovat nebo rušit existující databázové věty. Zadané nebo opravené hodnoty je třeba potvrdit stiskem klávesy F10 nebo tlačítkem „Save“.

Po stisku tlačítka „Query“ se formulář přepne do režimu zadávání dotazu. Tento režim je signalizován textem „ENTER QUERY“ zobrazeným ve stavovém řádku. V tomto režimu můžete zadávat dotaz tím, že do jednotlivých polí databázových položek zadáte kritéria výběru databázových vět. Např. po zadání hodnoty „Jan“ do položky JMENO budou vybrány všechny věty, které mají v položce JMENO hodnotu „Jan“. Dalším stiskem tlačítka „Query“ se zadaný dotaz provede. Prázdný dotaz vybere všechny věty z databázové tabulky. Vybrané databázové věty se zobrazí v editačních polích jednotlivých databázových položek. Zároveň se formulář přepne zpět do editačního režimu.

Tlačítka „<<“, „<“, „>“ a „>>“ slouží pro listování v seznamu vybraných databázových vět. Tlačítko „Help“ zobrazí jednoduchou nápovědu.

Po stisku tlačítka „Print Report“ se spustí aplikace „Reports RunTime“, která podle předem vytvořené a přeložené definice sestavy uložené v souboru „ex_rep.rep“ vygeneruje výstupní sestavu. Cílové zařízení, kam se sestava bude tisknout je „Screen“, tedy obrazovka. Je možné je změnit. Po stisku tlačítka „Run Report“ se sestava vytiskne. Bohužel, zatím zůstává po tisku sestavy neukončená aplikace „Reports Server“, kterou je nutné ukončit ručně pomocí funkce „Action“ -> „Quit“.

Tlačítko „Ukaz Dialog“ zobrazí dialogové okno. V jeho levé části jsou přepínací pole „Jak se mas“ a pravé části zaškrtačací pole „Co delas“. Jejich hodnoty je možné měnit standardním způsobem. Stiskem tlačítka „Help“ se zobrazí jednoduchá nápověda, po stisku tlačítka „Ok“ nebo „Cancel“ je dialogové okno uzavřeno.

Tlačítko „Kontrola“ zobrazí kontrolní okno. V jeho pravé části je sloupec zobrazovacích objektů. Objekty vedle nápisů „Nedelam Nic :“ a „Delam, ze delam :“ nezobrazují žádnou hodnotu. Objekty vedle nápisů „Pracuji :“ a „Hodnota Checkboxu Nic“ zobrazují stav, neboli hodnotu, zaškrtačacích objektů „Pracuji, co jineho“ a „No prece nic“ z dialogového okna. Hodnoty těchto zobrazovacích objektů se nastavují při zavírání dialogového okna, resp. při změně zaškrtačacího pole v dialogovém okně. Proto, pokud nebude nejprve otevřeno dialogové okno, resp. změněna hodnota zaškrtačacího pole, nebudou tyto zobrazovací objekty zobrazovat žádnou hodnotu. Tento stav je způsoben použitím přiřazovacích příkazů v různých trigrech z důvodu ukázat možnosti těchto trigrů bez ohledu na praktičnost nebo nepraktičnost tohoto řešení. Jak je podrobněji uvedeno v komentářích v jednotlivých trigrech, je samozřejmě možné tento stav změnit a formulář přeprogramovat jiným způsobem (viz. první odstavec kapitoly „Funkce demonstrační aplikace“). Zobrazovací objekt vedle textu „Hodnota radio grupy“ zobrazuje stav přepínacího pole z dialogového okna. Platí pro něj to, co pro zobrazování stavu zaškrtačacích polí. Vedle textu „Hodnota vracena alertem“ se, pokud bylo upozornění vyvoláno, zobrazí hodnota vrácená jako výsledek zobrazení upozornění. A konečně vedle nápisu „Zadany text“ se zobrazí text zadaný v editovacím poli v základním okně, pokud samozřejmě byl nějaký zadán. Tlačítko „Ok“ kontrolní okno uzavře.

Tlačítko „Alert“ zobrazí upozornění, což je okénko s definovaným textem a tlačítky.

Vlevo od tohoto tlačítka je editační pole. Text zde zadaný se zobrazí v kontrolním okně.

Popis menu demonstračního příkladu

Menu v modulu menu_ex_menu uloženém ve stejnojmenném souboru neprovádí žádné akce, pouze mohou zobrazit zprávu ve stavovém řádku. Slouží pouze k demonstraci možností systému menu.

Funkce „To je konec“ nebo „Konec“ ukončí běh demonstrační aplikace.

Popis demonstrační výstupní sestavy

Datový model sestavy obsahuje dotaz Q_1, který vybírá všechna data z tabulky novacekv.prac, a který tak definuje skupinu dat G_1. Dále jsou v něm definovány 4 objekty obsahující počty vět a součty platů na stránce a v celé sestavě.

Vzhled sestavy je tvořen několika hlavními objekty.

Rámec M_RepBody tvoří tělo sestavy. Obsahuje opakovací rámec R_1, který tiskne data ze skupiny G_1 vybraná dotazem Q_1, tedy všechna data z tabulky novacekv.prac. Protože množství těchto dat není předem známo, má rámec M_RepBody nastaven vertikální rozměr (Vertical Sizing) na Variable, aby mohlo pružně reagovat na množství dat a tedy velikost opakovacího rámce R_1.

Rámec M_RepBody dále obsahuje rámec M_Reptail, který tiskne patičku stránky obsahující mezisoučty a mezipočty na konci každé stránky. Jeho parametr „Print Condition“ je nastaven na „All“, „Anchoring“. To znamená, že se bude tisknout při každém výskytu otcovského objektu, tedy při každém výskytu rámce M_RepBody. Pokud bude dat tištěných v opakovacím rámci R_1 tolik, že se nevejdou na jednu stránku, bude tisk rámce M_RepBody pokračovat na další stránce a na jejím konci se opět vytiskne rámec M_Reptail.

Způsob navázání vazeb otec - syn mezi objekty R_1 a M_PgTail, M_PgTail a M_RepBody zajistí, že konec objektu R_1 nepřesáhne přes začátek objektu M_PgTail a konec objektu M_PgTail nepřesáhne přes konec objektu M_RepBody. Uspořádání objektů uvnitř rámce M_RepBody zároveň zajistí, že na každé stránce se nejprve pomocí opakovacího rámce R_1 tiskne seznam vět z tabulky

prac a na konci každé stránky se pomocí rámce M_PgTail vytiskne patička stránky.

Rámcem M_PgTitle má parametr „Print Condition“ nastaven na „All“, „Anchoring“. To znamená, že se bude tisknout při každém výskytu otcovského objektu, tedy při každém výskytu rámce M_RepBody. Jeho umístění v definici sestavy a způsob navázání explicitního vztahu otec - syn mezi jím a rámcem M_RepBody zajišťuje, že se na začátku každé stránky vytiskne nejprve rámcem M_PgTitle tak, že jeho konec nepřesáhne přes začátek rámce M_RepBody, a potom se bude tisknout rámcem M_RepBody. To znamená, že se takto tiskne hlavička každé stránky.

Rámcem M_Reptail má parametr „Print Condition“ nastaven na „Last“, „Anchoring“. Nemá nastaven žádný explicitní vztah otec - syn, jeho otcovským objektem je proto implicitně určen objekt „Report“. Vytiskne se proto na konci celé sestavy jako patička sestavy.

Poznámka

Pokud selhaly všechny možnosti a postupy a odpověď nebyla nalezena ani v tomto textu, je načase prostudovat manuál nebo dokumentaci.

Vytvořil Mgr. Vladimír Nováček
Poslední změna 13. října 1998