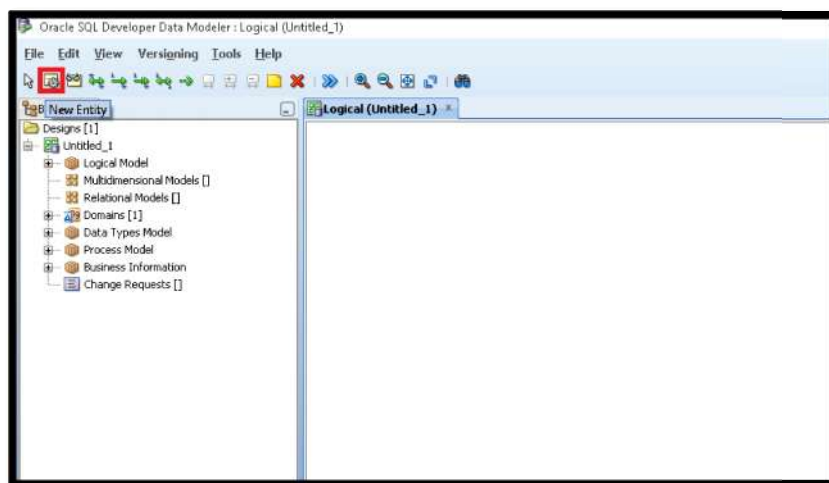


Tvorba logického a fyzického dátového modelu relačnej databázy pomocou nástrojov od firmy Oracle – výukový tutoriál

Prvým krokom pri vytvorení relačnej databázy je návrh jej štruktúry a realizácia logického dátového modelu. Tento model môžeme vytvoriť buď pomocou Oracle SQL Developer Data Modeler, alebo priamo v SQL Developeri.

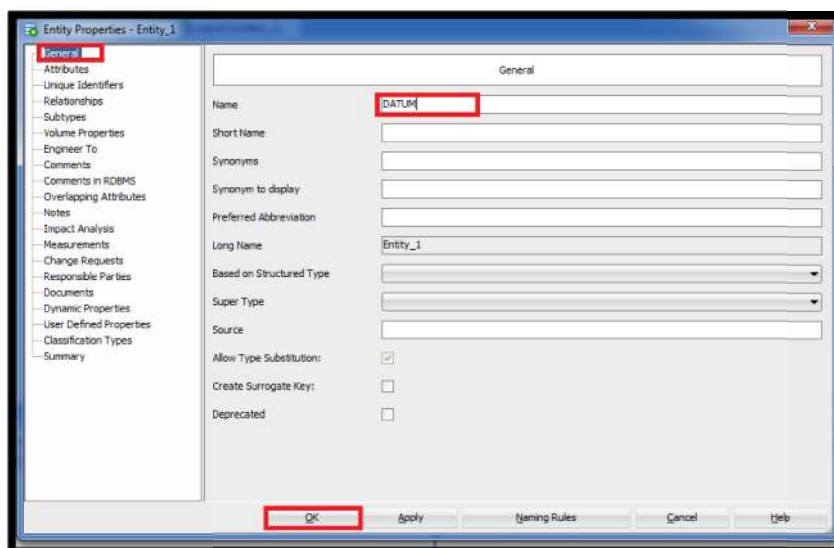
Realizácia logického dátového modelu

Logický dátový model definuje štruktúru dát v relačnej databáze, ako aj vzťahy medzi nimi. Logický model pozostáva z entít, ktoré obsahujú atribúty bližšie špecifikujúce konkrétnu entitu. Prvým krokom je teda vytvorenie novej entity (Obr. 1), keď po kliknutí na ikonku novej entity vytvoríme novú entitu kliknutím do plochy zobrazenej pod záložkou Logical.



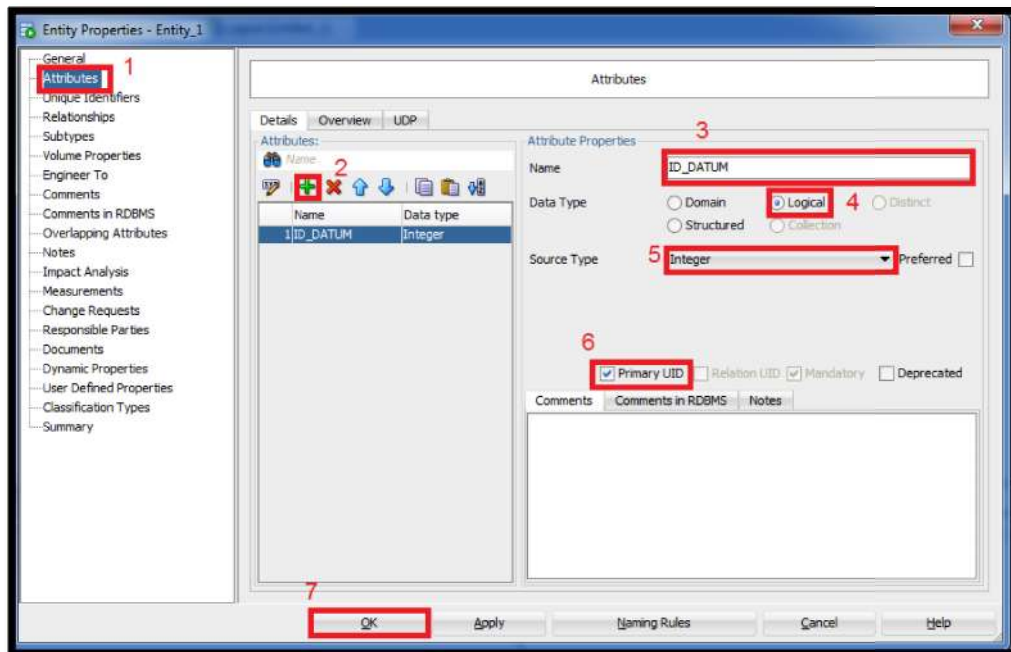
Obr. 1: Vytvorenie entity

Po vytvorení entity zadáme v záložke General názov danej entity (Obr. 2). V záložke Attributes potom špecifikujeme atribúty danej entity.



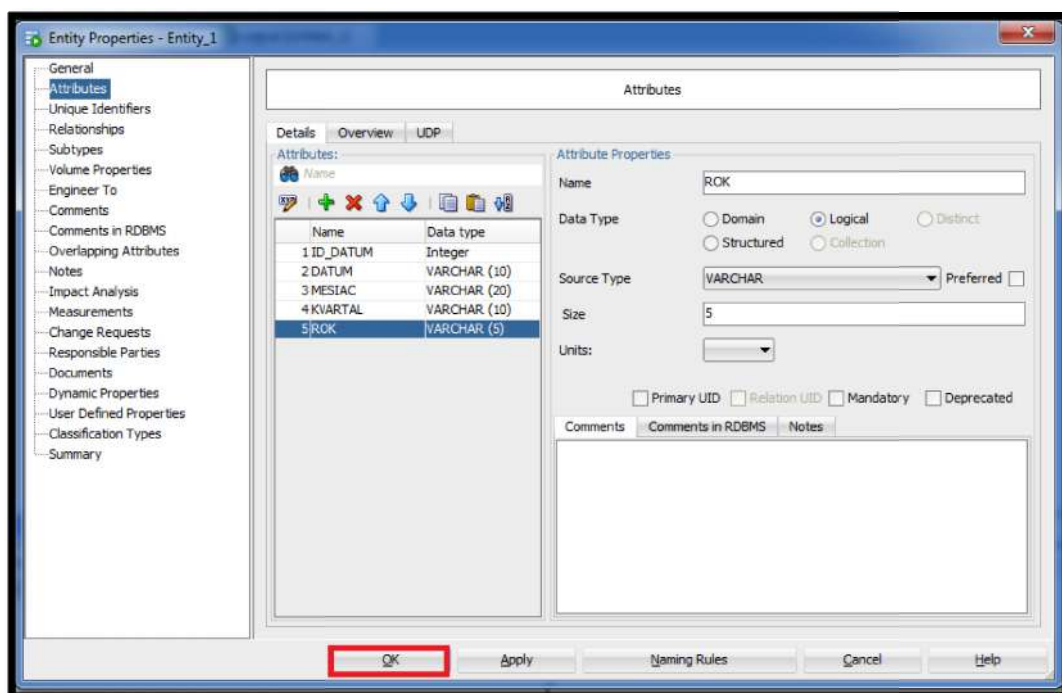
Obr. 2: Názov entity - záložka General

Každá entita by mala obsahovať svoj primárny unikátny jednoznačný identifikátor – UID. Postup pri vytváraní tohto atribútu je zobrazený na Obr. 3. Pokiaľ niektorý z atribútov entity označíme ako primárny UID, automaticky sa tento atribút stáva povinným a pri napĺňaní údajov do tabuľky nemôže táto hodnota ostať prázdna. Pri ostatných atribútoch je povinnosť vyplnenia daného atribútu voliteľná, dá sa zvoliť kliknutím na checkbox Mandatory (povinný).



Obr. 3: Tvorba primárneho jednoznačného identifikátora

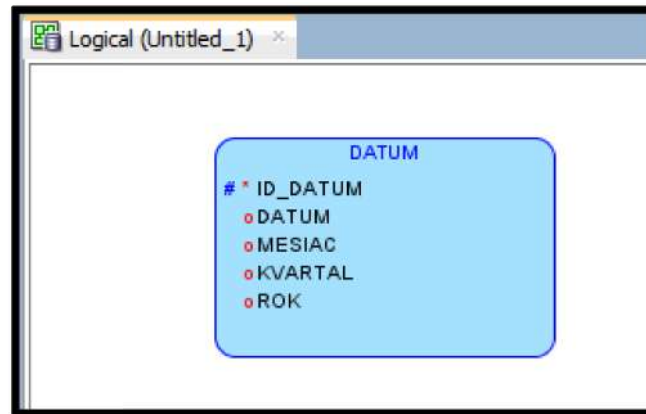
Po vytvorení primárneho UID vytvoríme aj ostatné atribúty danej entity. Finálnu entitu DATUM môžete vidieť na Obr. 4.



Obr. 4: Finálna entita DATUM

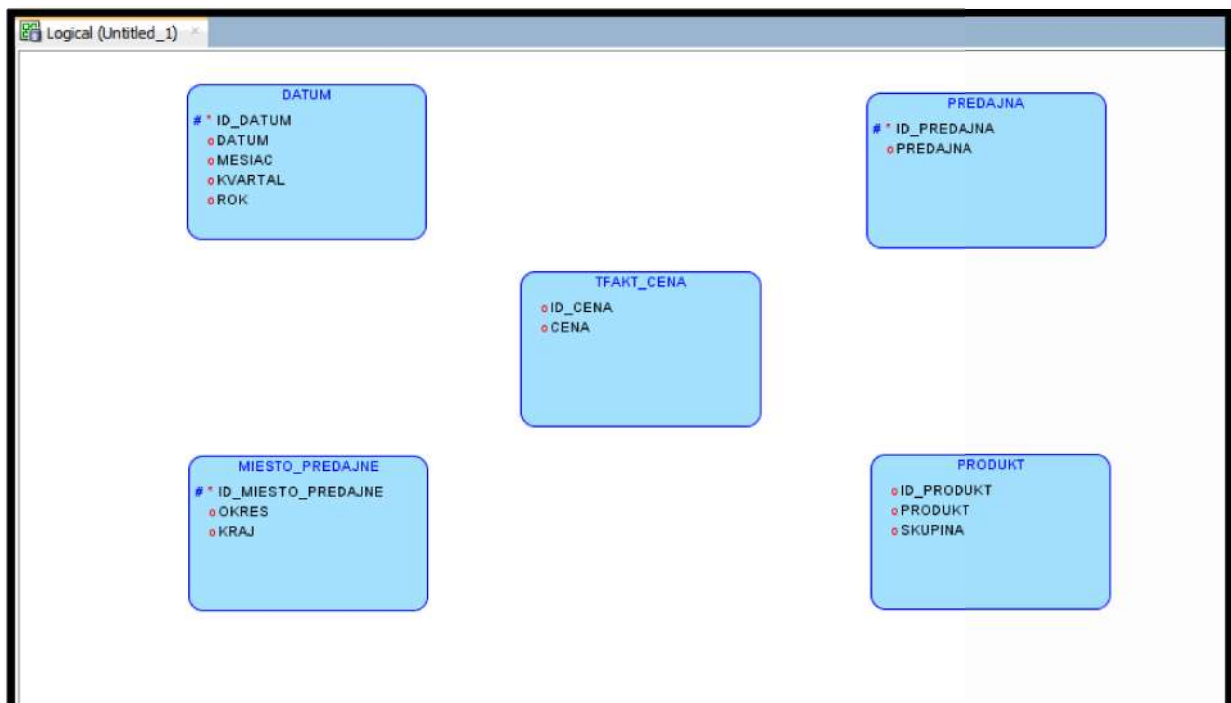
Po potvrdení všetkých parametrov sa v pracovnom priestore zobrazí vytvorená entita DATUM (Obr. 5). Symboly pri tejto entite:

- # - primárny unikátny identifikátor,
- * - povinný atribút,
- o – nepovinný atribút.



Obr. 5: Entita DATUM v pracovnom priestore

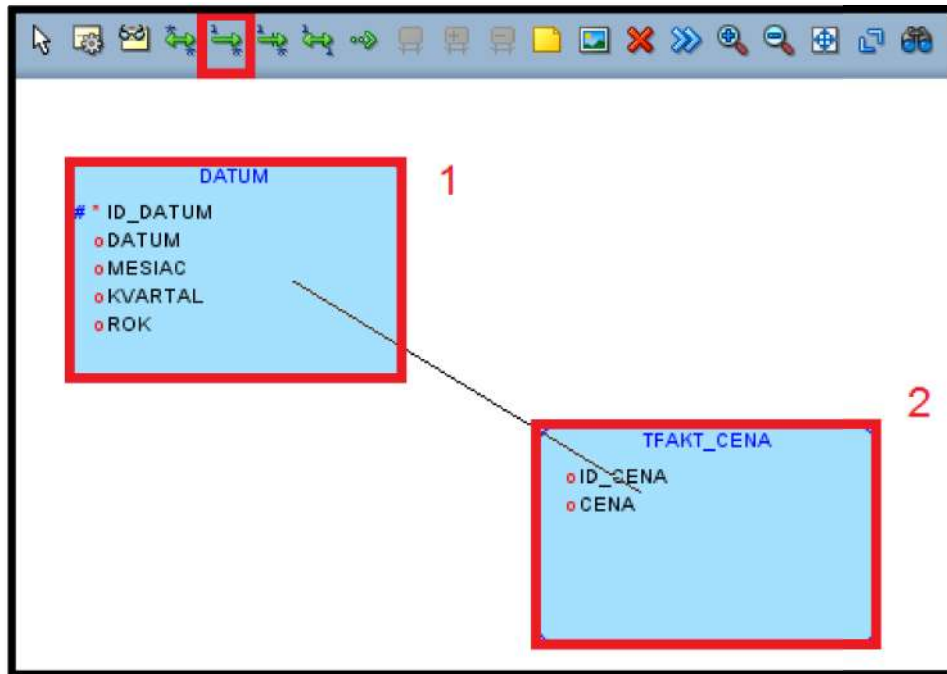
Analogicky vytvoríme aj ďalšie entity logického dátového modelu. Do tabuľky TFAKT_CENA nepridávame ako atribúty cudzie kľúče odkazujúce na tabuľky jednotlivých dimenzií. Stĺpce pre cudzie kľúče budú vygenerované do fyzického modelu automaticky na základe väzieb medzi jednotlivými entitami. Logický model bez väzieb medzi entitami je zobrazený na Obr. 6.



Obr. 6: Entity logického modelu - bez väzieb

Po vytvorení entít je potrebné ešte tieto entity prepojiť väzbami. Keďže tabuľka faktov obsahuje cudzie kľúče odvolávajúce sa na primárne kľúče v tabuľkách jednotlivých dimenzií (DATUM,

MIESTO_PREDAJNE, PREDAJNA, PRODUKT), musíme vytvoriť väzby medzi tabuľkami dimenzií a tabuľkou faktov. Keďže na 1 záznam v tabuľke príslušnej dimenzie sa môže odkazovať viac záznamov z tabuľky faktov, volíme väzbu 1:n (najprv klikneme na príslušnú dimenziu, potom na entitu pre tabuľku faktov). Postup tvorby väzby je ilustrovaný na Obr. 7.



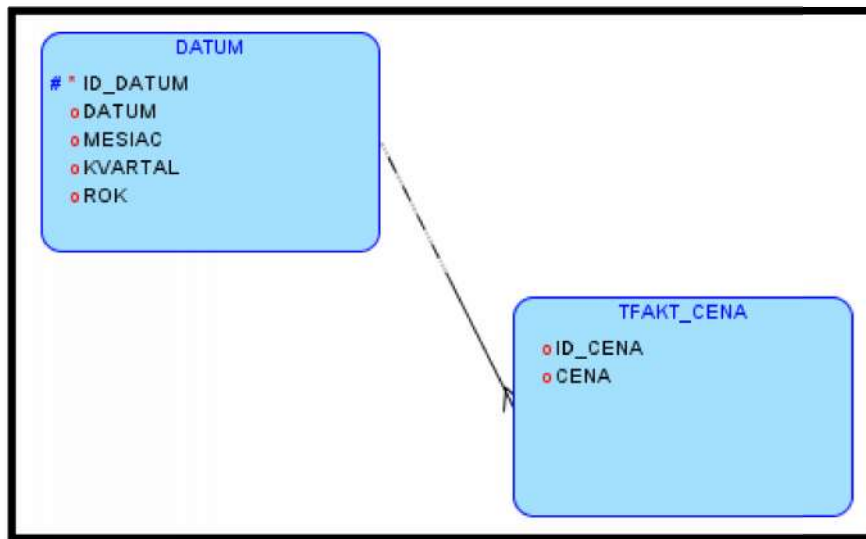
Obr. 7: Tvorba väzby

Po vytvorení väzby sa nám zobrazia možnosti pre vytváranú väzbu (Obr. 8).

The screenshot shows the 'Relation Properties - Relation_1' dialog box. The 'General' tab is selected. The 'Name' field contains 'Relation_1'. The 'Use surrogate keys' checkbox is unchecked. The 'Source Cardinality' section shows 'Source' as 'DATUM', 'Source key' as 'DATUM.DATUM PK', 'Name on Source' as an empty field, 'Source Entity Synonym' as 'DATUM', and 'Source to Target Cardinality' as '1-*'. The 'Source Optional' checkbox is checked, and 'Transferable' is also checked. The 'Target Cardinality' section shows 'Target' as 'TFAKT_CENA', 'Target key' as an empty field, 'Name on Target' as an empty field, 'Target Entity Synonym' as 'TFAKT_CENA', and 'Target to Source Cardinality' as '*-1'. The 'Target Optional' checkbox is unchecked, and 'Transferable' is checked. The 'Dominant Role' is set to 'None', 'Identifying' is unchecked, and 'Delete Rule' is set to 'NO ACTION'. The 'In Arc' checkbox is unchecked. The 'OK' button at the bottom is highlighted with a red box.

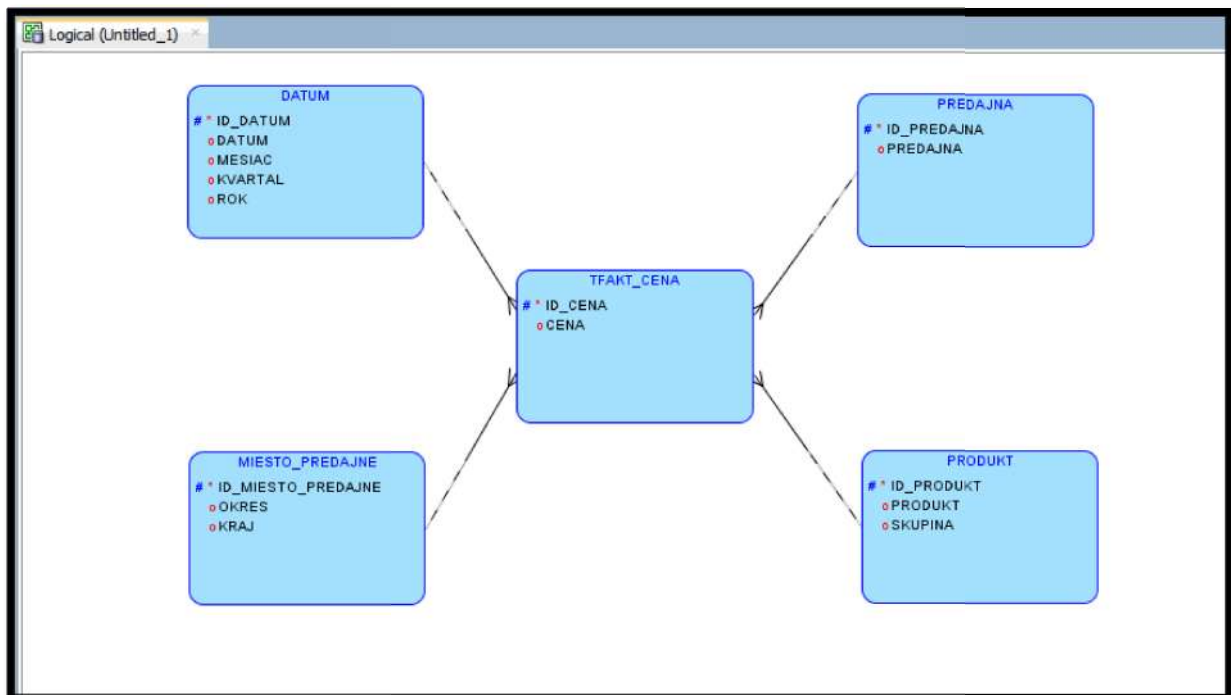
Obr. 8: Možnosti väzby

Po potvrdení v dátovom modeli pribudne čiara reprezentujúca väzbu medzi entitou DATUM a entitou TFAKT_CENA(Obr. 9).



Obr. 9: Väzba medzi entitami

Analogicky vytvoríme aj väzby medzi ostatnými entitami a entitou tabuľky faktov. Finálny logický model relačnej databázy je zobrazený na Obr. 10.

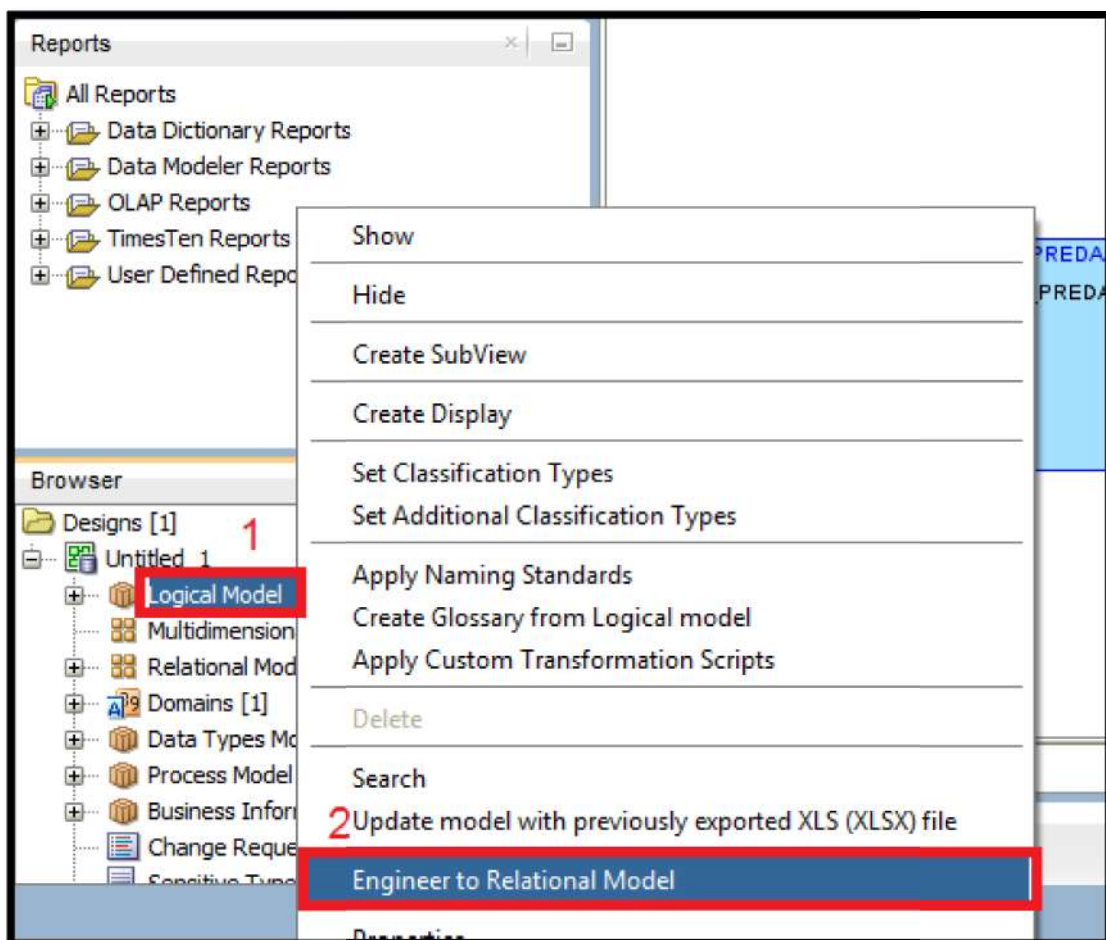


Obr. 10: Finálny logický model

Transformácia logického na fyzický model

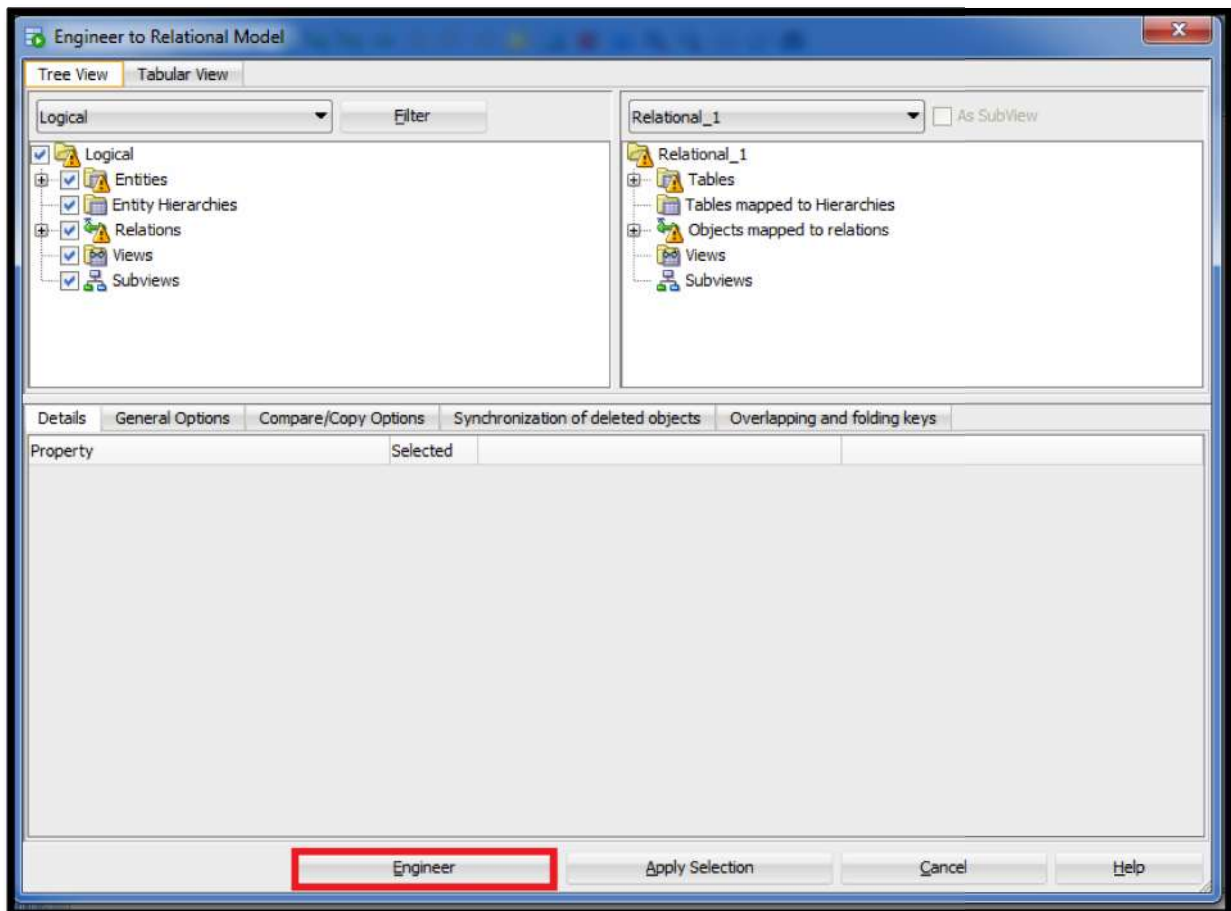
Kým logický model reprezentuje logickú reprezentáciu dát, entity a vzťahy medzi nimi, pre vytvorenie schémy v databáze je potrebná jeho transformácia na fyzickú úroveň, ktorá popisuje konkrétnu reprezentáciu dát v databáze (entity sú transformované na tabuľky, atribúty sa menia na stĺpce jednotlivých tabuliek...). V tomto modeli sa už zobrazujú aj dátové typy jednotlivých stĺpcov, vznikajú stĺpce pre cudzie kľúče, resp. pri väzbe m:n vznikajú prepájacie tabuľky.

Po vytvorení logického modelu jeho transformáciu na model fyzický začíname tak, že pravým tlačidlom klikneme na odkaz na logický model v okne Browser. Po kliknutí pravým tlačidlom zvolíme možnosť Engineer to Relational Model. Tento postup ilustruje Obr. 11.



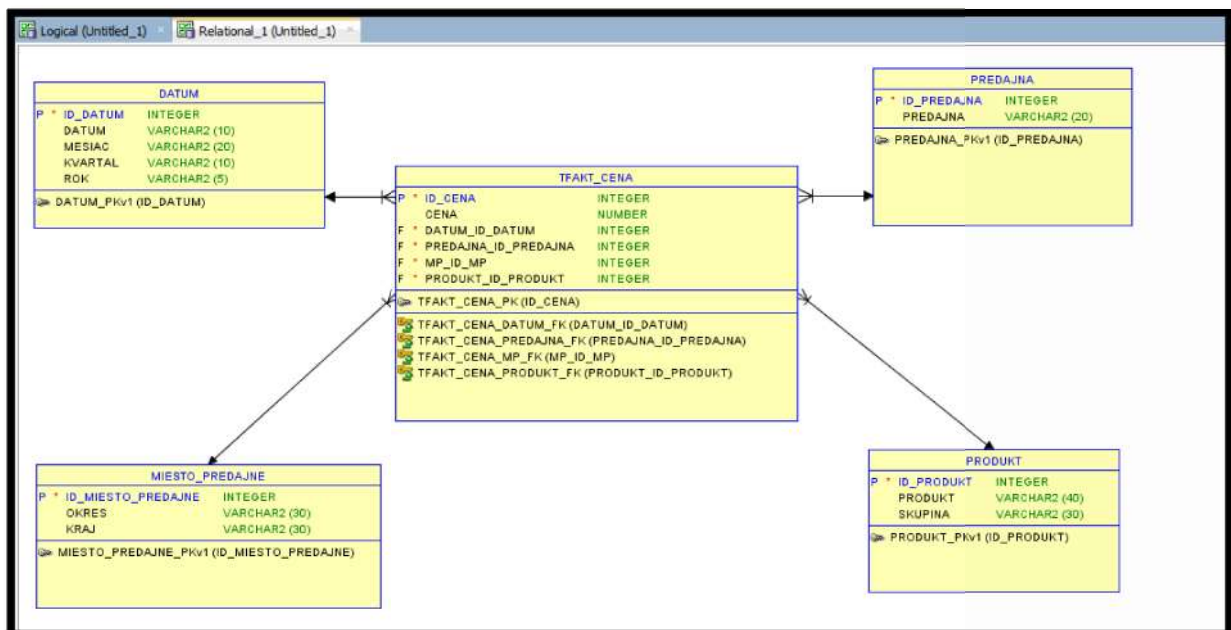
Obr. 11: Transformácia logického modelu na fyzický - krok 1

Po zvolení tejto možnosti sa zobrazí okno s nastaveniami možností transformácie. Po uskutočnení všetkých nastavení môžeme zvoliť možnosť Engineer (Obr. 12).



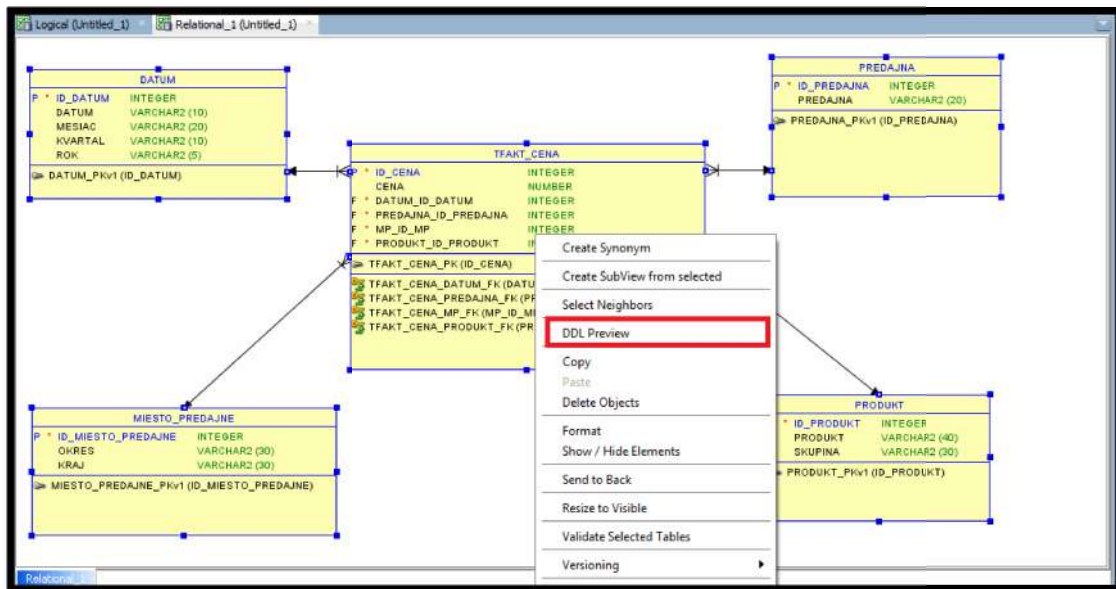
Obr. 12: Transformácia logického modelu na fyzický - krok 2

Po vykonaní transformácie sa zobrazí vygenerovaný fyzický model (Obr. 13).



Obr. 13: Fyzický model

Z fyzického modelu je už možné vygenerovať skript, pomocou ktorého implementujeme vytvorený dátový model do databázy. Tento skript vygenerujeme tak, že označíme všetky tabuľky a väzby vo fyzickom modeli, klikneme naň pravým tlačidlom a zvolíme možnosť DDL Preview (Obr. 14).



Obr. 14: Tvorba DDL Preview – skriptu pre importovanie tabuliek do databázy

Po zvolení tejto možnosti sa vygeneruje SQL skript (Obr. 15), ktorý vytvorí nami definovanú schému v databáze.

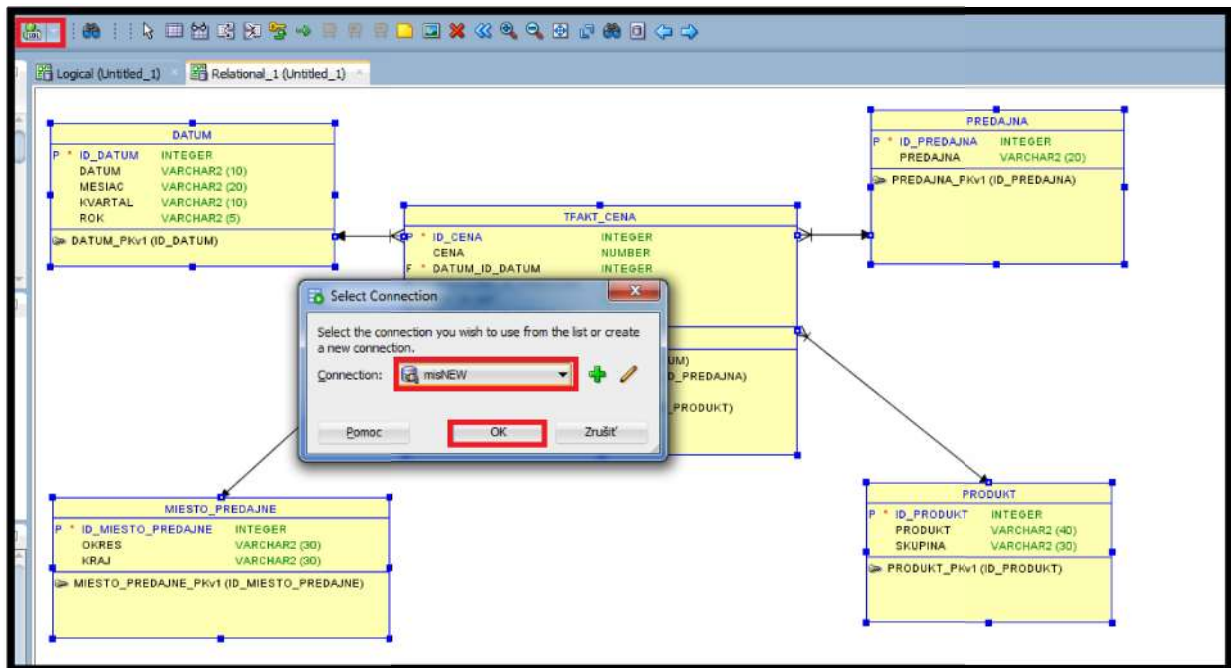
```

1 CREATE TABLE DATUM
2 (
3     ID_DATUM INTEGER NOT NULL ,
4     DATUM   VARCHAR2 (10) ,
5     MESIAC  VARCHAR2 (20) ,
6     KVARTAL VARCHAR2 (10) ,
7     ROK     VARCHAR2 (5)
8 );
9 ALTER TABLE DATUM ADD CONSTRAINT DATUM_PKV1 PRIMARY KEY ( ID_DATUM ) ;
10 CREATE TABLE MIESTO_PREDAJNE
11 (
12     ID_Miesto_Predajne INTEGER NOT NULL ,
13     OKRES              VARCHAR2 (30) ,
14     KRAJ              VARCHAR2 (30)
15 );
16 ALTER TABLE MIESTO_PREDAJNE ADD CONSTRAINT MIESTO_PREDAJNE_PKV1 PRIMARY KEY ( ID_Miesto_Predajne ) ;
17 CREATE TABLE PREDAJNA
18 (
19     ID_Predajna INTEGER NOT NULL ,
20     PREDAJNA   VARCHAR2 (20)
21 );
22 ALTER TABLE PREDAJNA ADD CONSTRAINT PREDAJNA_PKV1 PRIMARY KEY ( ID_Predajna ) ;
23 CREATE TABLE PRODUKT
24 (
25     ID_Produkt INTEGER NOT NULL ,
26     PRODUKT   VARCHAR2 (40) ,
27     SKUPINA   VARCHAR2 (30)
28 );
29 ALTER TABLE PRODUKT ADD CONSTRAINT PRODUKT_PKV1 PRIMARY KEY ( ID_Produkt ) ;
30 CREATE TABLE TFAKT_CENA
31 (
32     ID_CENA          INTEGER NOT NULL ,
33     CENA            NUMBER ,
34     DATUM_ID_DATUM  INTEGER NOT NULL ,
35     PREDAJNA_ID_PREDAJNA INTEGER NOT NULL ,
36     MP_ID_MP        INTEGER NOT NULL ,

```

Obr. 15: DDL Preview

Vygenerovaný skript spustíme v SQL Worksheete, ktorý si zobrazíme kliknutím na príslušnú ikonu v SQL Developeri a zvolením aktuálneho pripojenia do databázy, v ktorom momentálne pracujeme (Obr. 16).



Obr. 16: Zobrazenie SQL Worksheetu

Po zobrazení pracovného hárku sem vložíme celý skript, a spustíme jeho realizáciu tlačidlom Run Script (Obr. 17).

The screenshot shows the SQL Developer interface with the SQL script in the Worksheet. The script contains the following SQL statements:

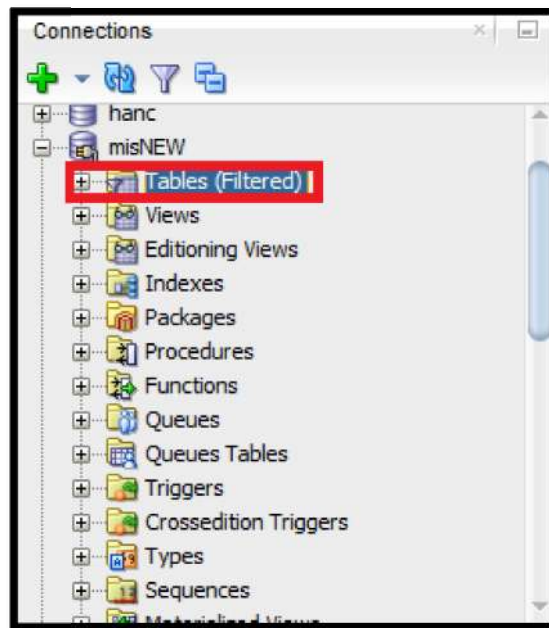
```

CREATE TABLE DATUM
(
  ID_DATUM INTEGER NOT NULL ,
  DATUM VARCHAR2 (10) ,
  MESIAC VARCHAR2 (20) ,
  KVARTAL VARCHAR2 (10) ,
  ROK VARCHAR2 (5)
);
ALTER TABLE DATUM ADD CONSTRAINT DATUM_PKv1 PRIMARY KEY ( ID_DATUM );
CREATE TABLE Miesto_Predajne
(
  ID_Miesto_Predajne INTEGER NOT NULL ,
  OKRES VARCHAR2 (30) ,
  KRAJ VARCHAR2 (30)
);
ALTER TABLE Miesto_Predajne ADD CONSTRAINT Miesto_Predajne_PKv1 PRIMARY KEY ( ID_Miesto_Predajne );
CREATE TABLE Predajna
(
  ID_Predajna INTEGER NOT NULL ,
  PREDAJNA VARCHAR2 (20)
);
ALTER TABLE Predajna ADD CONSTRAINT Predajna_PKv1 PRIMARY KEY ( ID_Predajna );
CREATE TABLE Produkt
(
  ID_PRODUKT INTEGER NOT NULL ,
  PRODUKT VARCHAR2 (40) ,
  SKUPINA VARCHAR2 (30)
);

```

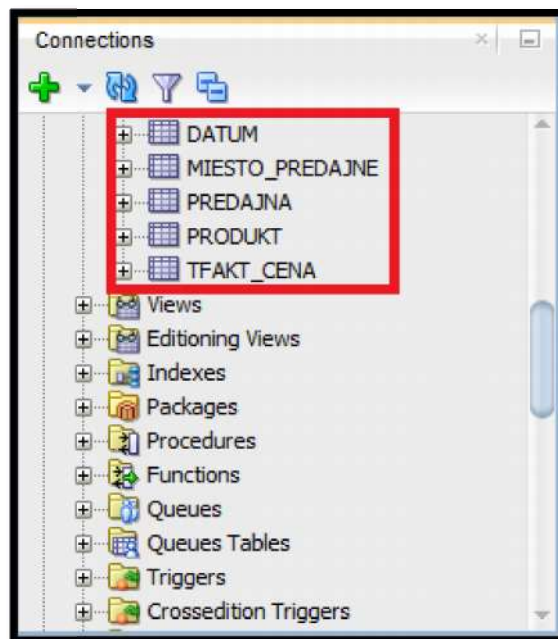
Obr. 17: Spustenie SQL skriptu

Pokiaľ realizácia skriptu prebehla bez chýb, v záložke Tables pri danom pripojení by sa mali zobraziť tabuľky, ktoré vytváraný dátový model obsahoval (Obr. 18, Obr. 19).



Obr. 18: Záložka TABLES

Tieto tabuľky sú dôkazom správne zrealizovaného procesu tvorby dátového modelu a môžu byť naplnené pripravenými dátami.



Obr. 19: Vytvorené tabuľky