**Data Definition Language (DDL) Statements used to Create the Store Schema**

Data Definition Language (DDL) statements are used to create users and tables, plus many other types of structures in the database. In this section, you’ll see the DDL statements used to create the store user and some of the tables.

**Creating a Database User:**

To create a user in the database, you use the CREATE USER statement. The simplified syntax for the CREATE USER statement is as follows:

**Syntax : CREATE USER user\_name IDENTIFIED BY password;**

where user\_name is the user name

password is the password for the user

**For example,** the following CREATE USER statement creates the store user with a password of store\_password:

**CREATE USER store IDENTIFIED BY store\_password;**

If you want the user to be able to work in the database, the user must be granted the necessary permissions to do that work. In the case of store, this user must be able to log onto the database (which requires the connect permission) and create items like database tables (which requires the resource permission). Permissions are granted by a privileged user (for example, the system user) using the GRANT statement.

The following example grants the connect and resource permissions to store:

**GRANT connect, resource TO store;**

**Examining the Store Tables**

In this section, you’ll learn how the tables for the store schema are created. Some of the information held in the store schema includes

* Customer details
* Types of products sold
* Product details
* A history of the products purchased by the customers
* Employees of the store
* Salary grades

The following tables are used to hold the information:

* **customers** holds the customer details.
* **product\_types** holds the types of products sold by the store.
* **products** holds the product details.
* **purchases** holds which products were purchased by which customers.
* **employees** holds the employee details.
* **salary\_grades** holds the salary grade details.

**The customers Table** The customers table holds the details of the customers. The following items are held in this table:

* First name
* Last name
* Date of birth (dob)
* Phone number

CREATE TABLE customers (customer\_id INTEGER CONSTRAINT customers\_pk PRIMARY KEY, first\_name VARCHAR2(10) NOT NULL,

last\_name VARCHAR2(10) NOT NULL,

dob DATE,

phone VARCHAR2(12) );

DESC customers;

SELECT \* FROM customers;

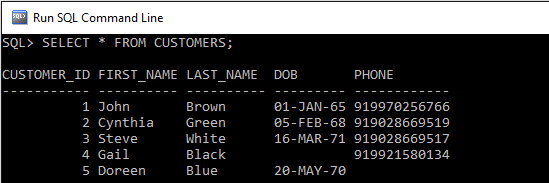
Insert into customers values(1, 'John', 'Brown','01-JAN-65', '919970256766');

Insert into customers values(2, 'Cynthia','Green','05-FEB-68', '919028669519');

Insert into customers values(3, 'Steve','White','16-MAR-71', '919028669517');

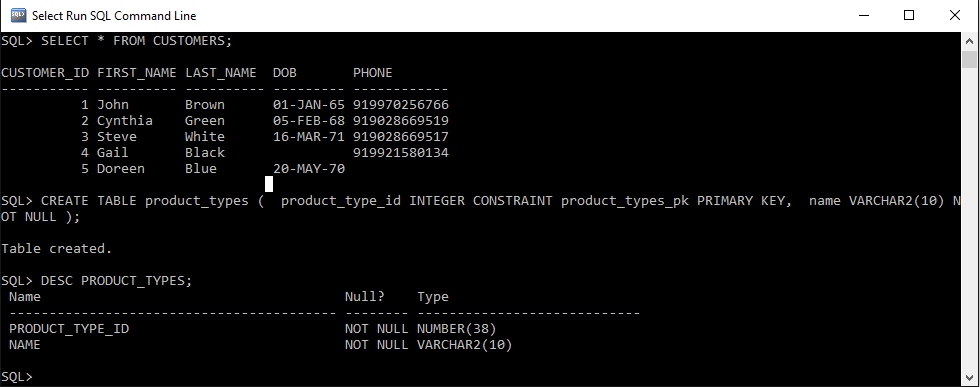
Insert into customers values(4, 'Gail','Black',null,'919921580134');

Insert into customers values(5, 'Doreen','Blue','20-MAY-70',NULL);



**The product\_types Table** The product\_types table holds the names of the product types sold by the store.

CREATE TABLE product\_types ( product\_type\_id INTEGER CONSTRAINT product\_types\_pk PRIMARY KEY, name VARCHAR2(10) NOT NULL );



SELECT \* FROM PRODUCT\_TYPES;

INSERT INTO PRODUCT\_TYPES VALUES(1,'Book');

INSERT INTO PRODUCT\_TYPES VALUES(2,'Video');

INSERT INTO PRODUCT\_TYPES VALUES(3,'DVD');

INSERT INTO PRODUCT\_TYPES VALUES(4,'CD');

INSERT INTO PRODUCT\_TYPES VALUES(5,'Magazine');

**The products Table** The products table holds the products sold by the store. The following pieces of information are held for each product:

Product type, Name, Description, Price

CREATE TABLE products ( product\_id INTEGER CONSTRAINT products\_pk PRIMARY KEY, product\_type\_id INTEGER CONSTRAINT products\_fk\_product\_types REFERENCES product\_types(product\_type\_id), name VARCHAR2(30) NOT NULL, description VARCHAR2(50), price NUMBER(5, 2) );

INSERT INTO PRODUCTS VALUES(1,1,'Modern Science','A description of modern science',19.95);

INSERT INTO PRODUCTS VALUES(2,1,'Chemistry','Introduction to Chemistry',30);

INSERT INTO PRODUCTS VALUES(3,2,'Super Nova','A star of explodes',25.99);

INSERT INTO PRODUCTS VALUES(4,2,'Tank War','Action movie about future war',13.95);

SELECT \* FROM PRODUCTS;

PRODUCT\_ID PRODUCT\_TYPE\_ID NAME DESCRIPTION PRICE

---------- --------------- ------------------------------ -------------------------------------------------- ----------

1 1 Modern Science A description of modern science 19.95

2 1 Chemistry Introduction to Chemistry 30

3 2 Super Nova A star of explodes 25.99

4 2 Tank War Action movie about future war 13.95

**The purchases Table** The purchases table holds the purchases made by a customer. For each purchase made by a customer, the following information is held:

Product ID, Customer ID, Number of units of the product that were purchased by the customer

CREATE TABLE purchases ( product\_id INTEGER CONSTRAINT purchases\_fk\_products REFERENCES products(product\_id), customer\_id INTEGER CONSTRAINT purchases\_fk\_customers REFERENCES customers(customer\_id), quantity INTEGER NOT NULL, CONSTRAINT purchases\_pk PRIMARY KEY (product\_id, customer\_id) );

INSERT INTO PURCHASES VALUES(1,1,1);

INSERT INTO PURCHASES VALUES(2,1,3);

INSERT INTO PURCHASES VALUES(1,4,1);

INSERT INTO PURCHASES VALUES(2,2,1);

INSERT INTO PURCHASES VALUES(1,3,1);

**OR**

INSERT INTO PURCHASES VALUES(&PID,&CID,&QTY);

SELECT \* FROM PURCHASES;

PRODUCT\_ID CUSTOMER\_ID QUANTITY

---------- ----------- ----------

1 1 1

2 1 3

1 4 1

2 2 1

1 3 1

**The employees Table** The employees table holds the details of the employees. The following information is held in the table:

* Employee ID
* The ID of the employee’s manager (if applicable)
* First name
* Last name
* Title
* Salary

CREATE TABLE employees

(

employee\_id INTEGER CONSTRAINT employees\_pk PRIMARY KEY,

manager\_id INTEGER,

first\_name VARCHAR2(10) NOT NULL,

last\_name VARCHAR2(10) NOT NULL,

title VARCHAR2(20),

salary NUMBER(6, 0)

);

INSERT INTO EMPLOYEES VALUES(1,NULL,'James','Smith','CEO',800000);

employee\_id manager\_id first\_name last\_name title salary

----------- ---------- ---------- ---------- ------------- ---------

1 James Smith CEO 800000

2 1 Ron Johnson Sales Manager 600000

3 2 Fred Hobbs Salesperson 150000

4 2 Susan Jones Salesperson 500000

**The salary\_grades Table** The salary\_grades table holds the different salary grades available to employees. The following information is held:

Salary grade ID, Low salary boundary for the grade, High salary boundary for the grade

CREATE TABLE salary\_grades

(

salary\_grade\_id INTEGER CONSTRAINT salary\_grade\_pk PRIMARY KEY,

low\_salary NUMBER(6, 0),

high\_salary NUMBER(6, 0)

);

INSERT INTO SALARY\_GRADES VALUES(1,1,250000);

salary\_grade\_id low\_salary high\_salary

--------------- ---------- ----------

1 1 250000

2 250001 500000

3 500001 750000

4 750001 999999