Marc Holman

CIS 2720 SQL I:NET

TERM PROJECT

11 / 25 / 2019

**Part I Deliverables:**

**--1 Scripts of Table Creation:**

/\* create table customers01 \*/

**CREATE TABLE customers01**

**(**

**customerid VARCHAR2(5),**

**firstname VARCHAR2(10) NOT NULL,**

**lastname VARCHAR2(10) NOT NULL,**

**city VARCHAR2(12),**

**st VARCHAR2(2),**

**zip VARCHAR2(5),**

**CONSTRAINT customers01\_customerid\_pk PRIMARY KEY(customerid)**

**);**

/\* create indices for customers01 \*/

**CREATE INDEX customers01\_zip\_idx**

**ON customers01(zip);**

**CREATE INDEX customer\_name\_idx**

**ON customers01 (lastname, firstname);**

/\* create table products01 \*/

**CREATE TABLE products01**

**(**

**productid VARCHAR2(5),**

**productname VARCHAR2(20) NOT NULL,**

**msrp NUMBER(8,2) NOT NULL,**

**category VARCHAR2(20) NOT NULL,**

**CONSTRAINT products01\_productid\_pk PRIMARY KEY(productid)**

**);**

/\* indices for products01 \*/

**CREATE INDEX products01\_msrp\_idx**

**ON products01(msrp);**

**CREATE INDEX products01\_prodname\_idx**

**ON products01(productname);**

/\* insert data from project\_file into customers01 and products01 \*/

**INSERT INTO customers01**

**SELECT DISTINCT customerid, firstname, lastname, city, state, zip**

**FROM project\_file;**

**INSERT INTO products01**

**SELECT DISTINCT productid, productname, msrp, category**

**FROM project\_file;**

/\* set up sequences for customers01 and products01 \*/

**CREATE SEQUENCE SEQ\_CUSTOMERID**

**INCREMENT BY 1**

**START WITH 120**

**MAXVALUE 999 /\* 3 digit id to spec. \*/**

**MINVALUE 1**

**CYCLE;**

**CREATE SEQUENCE SEQ\_PRODUCTID**

**INCREMENT BY 1**

**START WITH 210**

**MAXVALUE 999**

**MINVALUE 1**

**CYCLE;**

/\* insert 2 hypothetical records into customers01 and products01 \*/

**INSERT INTO customers01**

**(customerid, firstname, lastname, city, st, zip)**

**VALUES**

**(SEQ\_CUSTOMERID.NEXTVAL, 'Marc', 'Holman', 'Lombard', 'IL', '60148');**

**INSERT INTO customers01**

**(customerid, firstname, lastname, city, st, zip)**

**VALUES**

**(SEQ\_CUSTOMERID.NEXTVAL, 'Ronald', 'Reagan', 'Chicago', 'IL', '53467');**

**INSERT INTO products01**

**(productid, productname, msrp, category)**

**VALUES(SEQ\_PRODUCTID.NEXTVAL, 'Television', 499.99, 'LivingRoom');**

**INSERT INTO products01**

**(productid, productname, msrp, category)**

**VALUES(SEQ\_PRODUCTID.NEXTVAL, 'Fax', 29.99, 'Office');**

/\* create table orders with initial constraints \*/

**CREATE TABLE ORDERS01**

**(**

**orderid VARCHAR2(5)**

**,saleid NUMBER (5)**

**,customerid VARCHAR2(5)**

**,productid VARCHAR2(5)**

**,saleprice NUMBER(8,2)**

**,saledate DATE**

**,quantity INTEGER DEFAULT 1**

**,CONSTRAINT saleid\_fk FOREIGN KEY (saleid) REFERENCES project\_file(saleid)**

**,CONSTRAINT custid\_fk FOREIGN KEY (customerid) REFERENCES customers01(customerid)**

**,CONSTRAINT productid\_fk FOREIGN KEY (productid) REFERENCES products01(productid)**

**);**

/\* set up composite primary key on orders01 \*/

**ALTER TABLE orders01**

**ADD CONSTRAINT orders01\_composite\_pk PRIMARY KEY(customerid, productid, saleid);**

/\* add shipdate virtual column to orders01 \*/

**ALTER TABLE orders01**

**ADD shipdate DATE as (saledate + 2);**

/\* insert data into orders01 table from project\_file \*/

**INSERT INTO orders01(customerid, productid, saleid, saleprice, saledate)**

**SELECT DISTINCT customerid, productid, saleid, saleprice, to\_date(saledate, 'MM/DD/YYYY')**

**FROM project\_file;**

/\* create 5 digit sequence for orders01.orderid: seq\_orderid

and for orders01.saleid: seq\_saleid \*/

**CREATE SEQUENCE SEQ\_ORDERID**

**INCREMENT BY 1**

**START WITH 100**

**MAXVALUE 99999**

**MINVALUE 1**

**CYCLE;**

**CREATE SEQUENCE SEQ\_SALEID**

**INCREMENT BY 1**

**START WITH 360**

**MAXVALUE 99999**

**MINVALUE 1**

**CYCLE;**

/\* use sequence seq\_orderid to populate orders01.orderid \*/

**UPDATE orders01**

**SET orderid = SEQ\_ORDERID.NEXTVAL;**

/\* drop composite primary key constraint on orders01 and saleid fk constraint \*/

**ALTER TABLE orders01**

**DROP CONSTRAINT orders01\_composite\_pk;**

/\* drop saleid\_fk constraint - project\_file now out of the picture \*/

**ALTER TABLE orders01**

**DROP CONSTRAINT saleid\_fk;**

/\* add constraint to set orderid as primary key for orders01 table \*/

**ALTER TABLE orders01**

**ADD CONSTRAINT orders01\_orderid\_pk PRIMARY KEY(orderid);**

/\* Create indices for orders01 table \*/

**CREATE INDEX orders01\_saledate\_idx**

**ON orders01(saledate);**

**CREATE INDEX orders01\_shipdate\_idx**

**ON orders01(shipdate);**

/\* add 4 hypothetical but logical records to the orders01 table \*/

**INSERT INTO orders01**

**(orderid, saleid, customerid, productid, saleprice, saledate, quantity)**

**VALUES(SEQ\_ORDERID.NEXTVAL, SEQ\_SALEID.NEXTVAL, 160, 201, 150, '06-FEB-04', 2);**

**INSERT INTO orders01**

**(orderid, saleid, customerid, productid, saleprice, saledate, quantity)**

**VALUES(SEQ\_ORDERID.NEXTVAL, SEQ\_SALEID.NEXTVAL, 161, 201, 150, '07-FEB-04', 3);**

**INSERT INTO orders01**

**(orderid, saleid, customerid, productid, saleprice, saledate, quantity)**

**VALUES(SEQ\_ORDERID.NEXTVAL, SEQ\_SALEID.NEXTVAL, 106, 202, 175, '27-JUL-19', 1);**

**INSERT INTO orders01**

**(orderid, saleid, customerid, productid, saleprice, saledate, quantity)**

**VALUES(SEQ\_ORDERID.NEXTVAL, SEQ\_SALEID.NEXTVAL, 160, 203, 250, '10-OCT-19', 3);**

/\* commit above transactions \*/

**commit;**

**--2 Show All Constraints Set up:**

/\* This statement will show all existing constraints \*/

**SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, SEARCH\_CONDITION, STATUS**

**FROM USER\_CONSTRAINTS**

**WHERE TABLE\_NAME IN ('products01', 'customers01', 'orders01');**

**OUTPUT:**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE\_NAME,"CONSTRAINT\_NAME","CONSTRAINT\_TYPE","SEARCH\_CONDITION","STATUS" |  |  |  |
| CUSTOMERS01,"SYS\_C0011780","C","""FIRSTNAME"" | IS | NOT | NULL","ENABLED" |
| CUSTOMERS01,"SYS\_C0011781","C","""LASTNAME"" | IS | NOT | NULL","ENABLED" |
| CUSTOMERS01,"CUSTOMERS01\_CUSTOMERID\_PK","P","","ENABLED" |  |  |  |
| PRODUCTS01,"SYS\_C0011788","C","""MSRP"" | IS | NOT | NULL","ENABLED" |
| PRODUCTS01,"SYS\_C0011789","C","""CATEGORY"" | IS | NOT | NULL","ENABLED" |
| PRODUCTS01,"PRODUCTS01\_PRODUCTID\_PK","P","","ENABLED" |  |  |  |
| PRODUCTS01,"SYS\_C0011787","C","""PRODUCTNAME"" | IS | NOT | NULL","ENABLED" |
| ORDERS01,"CUSTID\_FK","R","","ENABLED" |  |  |  |
| ORDERS01,"PRODUCTID\_FK","R","","ENABLED" |  |  |  |
| ORDERS01,"ORDERS01\_ORDERID\_PK","P","","ENABLED" |  |  |  |

**--3** Entity Relationship Diagrams for orders01, products01, customers01.

Orders01 serves as a bridging entity between customers and products with one to many relationships existing between customers01 and order01 and products01 and orders.

Products01

productid pk

productname

msrp

category

Orders01

ordereid pk

saleid fk

customerid fk

productid fk

saleprice

saledate

quantity

Customers01

customerid pk

firstname

lastname

city

st

zip

**Part II Deliverables - 6 Queries:**

**--1** Return customerid, firstname, lastname of those customers who made no purchases during the final quarter of 2005:

**SELECT DISTINCT customerid, firstname, lastname**

**FROM customers01**

**JOIN orders01 USING(customerid)**

**WHERE saledate < '01-OCT-05';**

**Output:**

|  |
| --- |
| CUSTOMERID,"FIRSTNAME","LASTNAME" |
| 115,"Kelly","Street" |
| 117,"Holly","Raines" |
| 160,"Marc","Holman" |
| 109,"Billy","Costigan" |
| 106,"Irwin","Wade" |
| 116,"Tricia","Hill" |
| 161,"Ronald","Reagan" |
| 101,"John","Miller" |
| 105,"Mike","Horvath" |
| 108,"Frank","Costello" |
| 103,"Stan","Mellish" |
| 118,"Natalie","Woods" |
| 119,"Wendy","Hilton" |
| 104,"Adrian","Caparzo" |
| 114,"Kristy","Bryant" |
| 102,"Fred","Hammill" |
| 107,"George","Marshall" |

**--2** Return firstname, lastname, productname, and saleprice for all products sold in October of 2005 in order of high sale price to low:

**SELECT c.firstname, c.lastname, p.productname, o.saleprice, o.saledate**

**FROM customers01 c, products01 p, orders01 o**

**WHERE o.productid = p.productid**

**AND o.customerid = c.customerid**

**AND o.saledate > '01-OCT-05' AND o.saledate < '31-OCT-05'**

**ORDER BY saleprice DESC;**

**Output:**

|  |
| --- |
| FIRSTNAME,"LASTNAME","PRODUCTNAME","SALEPRICE","SALEDATE" |
| Mike,"Horvath","Refrigerator",1100,10-OCT-05 |
| Mike,"Horvath","Refrigerator",899,10-OCT-05 |
| Irwin,"Wade","VCR",150,11-OCT-05 |
| George,"Marshall","Microwave",87,12-OCT-05  **--3** return firstname, lastname, saleprice, msrp, and discount (difference between saleprice and msrp) for all sales, showing only the sales with saleprice equal to or less than msrp:  **SELECT c.firstname, c.lastname, TO\_CHAR(o.saleprice, '$999.99') "Saleprice", TO\_CHAR(p.msrp, '$999.99') "MSRP", TO\_CHAR((p.msrp - o.saleprice), '$999.99') "Discount"**  **FROM customers01 c**  **JOIN orders01 o USING(customerid)**  **JOIN products01 p USING(productid)**  **WHERE o.saleprice <= p.msrp;**  FIRSTNAME,"LASTNAME","Saleprice","MSRP","Discount"  Kristy,"Bryant"," $85.00"," $98.00"," $13.00"  Kristy,"Bryant"," $99.00"," $165.00"," $66.00"  Adrian,"Caparzo"," $80.00"," $85.00"," $5.00"  Frank,"Costello"," $198.00"," $200.00"," $2.00"  Billy,"Costigan"," $130.00"," $165.00"," $35.00"  Fred,"Hammill"," $97.00"," $98.00"," $1.00"  Tricia,"Hill"," $99.00"," $105.00"," $6.00"  Wendy,"Hilton"," $150.00"," $165.00"," $15.00"  Mike,"Horvath"," $899.00"," $900.00"," $1.00"  Mike,"Horvath"," $900.00"," $900.00"," $.00"  George,"Marshall"," $88.00"," $98.00"," $10.00"  George,"Marshall"," $87.00"," $98.00"," $11.00"  Stan,"Mellish"," $200.00"," $200.00"," $.00"  Holly,"Raines"," $87.00"," $105.00"," $18.00"  Kelly,"Street"," $104.00"," $165.00"," $61.00"  Irwin,"Wade"," $150.00"," $165.00"," $15.00" |

**--4** Return customerid, average saleprice for those customers who have purchased three or more products, format average saleprice with $ sign and 2 decimal and rank by average saleprice from high to low.

**SELECT customerid "ID", COUNT(customerid) "Num. Orders", SUM(quantity) "Total Qty.", TO\_CHAR(AVG(saleprice), '$9999.99') "Average Saleprice"**

**FROM orders01**

**GROUP BY customerid**

**HAVING SUM(quantity) >= 3**

**ORDER BY "Average Saleprice" DESC;**

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID,"Num. | Orders","Total | Qty.","Average | Saleprice" |
| 105,3,3," |  | $966.33" |  |
| 160,2,5," |  | $200.00" |  |
| 108,3,3," |  | $196.00" |  |
| 106,4,4," |  | $174.00" |  |
| 161,1,3," |  | $150.00" |  |
| 107,3,3," |  | $128.00" |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |  |  |  |   **--5** Randomly select 6 customers with customerid, lastname, and firstname and zip who bought products in office category.  **SELECT Distinct o.customerid, c.lastname, c.firstname, c.zip, p.category**  **FROM customers01 c, products01 p, orders01 o**  **WHERE c.customerid = o.customerid**  **AND p.productid = o.productid**  **AND p.category = 'Office'**  **ORDER BY dbms\_random.VALUE FETCH NEXT 6 ROWS ONLY;**  **Output:**   |  | | --- | | CUSTOMERID,"LASTNAME","FIRSTNAME","ZIP","CATEGORY" | | 160,"Holman","Marc","60148","Office" | | 107,"Marshall","George","34908","Office" | | 115,"Street","Kelly","57732","Office" | | 106,"Wade","Irwin","45902","Office" | | 108,"Costello","Frank","23905","Office" | | 116,"Hill","Tricia","46738","Office" |   **--6** Get customer count and sales by state in the order from high sales to low. Total customer count and sales value and label it ‘Total’ in the state field.  **SELECT st "State", COUNT(customerid) "Total Customers", TO\_CHAR(SUM(saleprice), '$9999.99') "Total Sales"**  **FROM orders01**  **JOIN customers01 USING(customerid)**  **GROUP BY st**  **ORDER BY "Total Sales" DESC;** |  |  |

**Output:**

|  |  |  |
| --- | --- | --- |
| State,"Total | Customers","Total | Sales" |
| IN,4,$2998.00 |  |  |
| CO,2,$1080.00 |  |  |
| KY,5,$846.00 |  |  |
| HI,3,$588.00 |  |  |
| IL,3,$550.00 |  |  |
| ND,3,$384.00 |  |  |
| ME,2,$369.00 |  |  |
| ID,2,$344.00 |  |  |
| WY,2,$314.00 |  |  |
| SC,2,$280.00 |  |  |
| NY,2,$260.00 |  |  |
| AK,2,$199.00 |  |  |
| FL,2,$184.00 |  |  |
| MS,2,$177.00 |  |  |