*Gabriel Nestor* ITI-2180 Foundations of Data and Database Management

**Problem A**

**CODE:**

CREATE TABLE CUSTOMERS\_ALT AS

SELECT CUSTOMER#, LASTNAME, FIRSTNAME, STATE, ZIP ZIPCODE

FROM CUSTOMERS;

SELECT \*

FROM CUSTOMERS\_ALT;

**OUTPUT:**

Table CUSTOMERS\_ALT created.

CUSTOMER# LASTNAME FIRSTNAME ST ZIPCO

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

1001 MORALES BONITA FL 32328

1002 THOMPSON RYAN CA 90404

1003 SMITH LEILA FL 32306

1004 PIERSON THOMAS ID 83707

1005 GIRARD CINDY WA 98115

1006 CRUZ MESHIA NY 12211

1007 GIANA TAMMY TX 78710

1008 JONES KENNETH WY 82003

1009 PEREZ JORGE CA 91510

1010 LUCAS JAKE GA 30314

1011 MCGOVERN REESE IL 60606

CUSTOMER# LASTNAME FIRSTNAME ST ZIPCO

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

1012 MCKENZIE WILLIAM MA 02110

1013 NGUYEN NICHOLAS FL 34711

1014 LEE JASMINE WY 82414

1015 SCHELL STEVE FL 33111

1016 DAUM MICHELL CA 91508

1017 NELSON BECCA MI 49006

1018 MONTIASA GREG GA 31206

1019 SMITH JENNIFER NJ 07962

1020 FALAH KENNETH NJ 08607

20 rows selected.

**Problem B**

**CODE:**

ALTER TABLE CUSTOMERS\_ALT

ADD CONSTRAINT CUSTOMERS\_ALT\_CUSTOMER#\_PK PRIMARY KEY (CUSTOMER#)

ADD CONSTRAINT TAX\_STATE\_FK FOREIGN KEY (STATE)

REFERENCES TAX (STATE);

**OUTPUT:**

Table CUSTOMERS\_ALT altered.

**Problem C**

**CODE:**

CREATE INDEX CUSTOMERS\_ALT\_IDX

ON CUSTOMERS\_ALT (ZIPCODE);

SELECT TABLE\_NAME, INDEX\_NAME, COLUMN\_NAME

FROM USER\_IND\_COLUMNS

WHERE TABLE\_NAME = 'CUSTOMERS\_ALT';

DROP INDEX CUSTOMERS\_ALT\_IDX;

**OUTPUT:**

Index CUSTOMERS\_ALT\_IDX created.

TABLE\_NAME INDEX\_NAME COLUMN\_NAME

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

CUSTOMERS\_ALT CUSTOMERS\_ALT\_CUSTOMER#\_PK CUSTOMER#

CUSTOMERS\_ALT CUSTOMERS\_ALT\_IDX ZIPCODE

Index CUSTOMERS\_ALT\_IDX dropped.

**Problem D**

**CODE:**

CREATE SEQUENCE ORDERS\_ORDER#\_SEQ

INCREMENT BY 2

START WITH 1022

NOCACHE NOCYCLE;

INSERT INTO ORDERS

VALUES (ORDERS\_ORDER#\_SEQ.NEXTVAL, NULL, SYSDATE, NULL, NULL, NULL, NULL, NULL, NULL);

INSERT INTO ORDERS

VALUES (ORDERS\_ORDER#\_SEQ.NEXTVAL, NULL, SYSDATE, NULL, NULL, NULL, NULL, NULL, NULL);

SELECT ORDER#

FROM ORDERS;

**OUTPUT:**

Sequence ORDERS\_ORDER#\_SEQ created.

1 row inserted.

1 row inserted.

ORDER#

----------

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

ORDER#

----------

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1022

ORDER#

----------

1024

23 rows selected.

**Problem E**

**CODE:**

SELECT TITLE, CATEGORY, RETAIL

FROM BOOKS

WHERE CATEGORY = 'COMPUTER' AND PUBID = 3 AND RETAIL >= 30;

**OUTPUT:**

TITLE CATEGORY RETAIL

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

DATABASE IMPLEMENTATION COMPUTER 55.95

HOLY GRAIL OF ORACLE COMPUTER 75.95

**Problem F**

**CODE:**

SELECT FIRSTNAME || ' ' || LASTNAME AS "Customers in Atlanta, GA"

FROM CUSTOMERS JOIN ORDERS USING (CUSTOMER#)

WHERE UPPER(SHIPCITY) = 'ATLANTA' AND UPPER(SHIPSTATE) = 'GA';

**OUTPUT:**

Customers in Atlanta, GA

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

JAKE LUCAS

JAKE LUCAS

**Problem G**

**CODE:**

SELECT TITLE, TO\_CHAR(RETAIL, '999.99') AS "EXISTING PRICE", TO\_CHAR((ROUND((RETAIL\*.05)+RETAIL, 0)-.01), '999.99') AS "PROPOSED PRICE"

FROM BOOKS

ORDER BY TITLE;

**OUTPUT:**

TITLE EXISTIN PROPOSE

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

BIG BEAR AND LITTLE DOVE 8.95 8.99

BODYBUILD IN 10 MINUTES A DAY 30.95 31.99

BUILDING A CAR WITH TOOTHPICKS 59.95 62.99

COOKING WITH MUSHROOMS 19.95 20.99

DATABASE IMPLEMENTATION 55.95 58.99

E-BUSINESS THE EASY WAY 54.50 56.99

HANDCRANKED COMPUTERS 25.00 25.99

HOLY GRAIL OF ORACLE 75.95 79.99

HOW TO GET FASTER PIZZA 29.95 30.99

HOW TO MANAGE THE MANAGER 31.95 33.99

PAINLESS CHILD-REARING 89.95 93.99

TITLE EXISTIN PROPOSE

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

REVENGE OF MICKEY 22.00 22.99

SHORTEST POEMS 39.95 41.99

THE WOK WAY TO COOK 28.75 29.99

14 rows selected.

**Problem H**

**CODE:**

SELECT CUSTOMER#, SUM(NUMBOOKS) AS "Ordered Books"

FROM CUSTOMERS JOIN (SELECT CUSTOMER#, ORDER#, SUM(QUANTITY) NUMBOOKS FROM ORDERS NATURAL JOIN ORDERITEMS GROUP BY CUSTOMER#, ORDER#) USING (CUSTOMER#)

GROUP BY CUSTOMER#

ORDER BY CUSTOMER#;

**OUTPUT:**

CUSTOMER# Ordered Books

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

1001 5

1003 2

1004 2

1005 3

1007 8

1008 1

1010 3

1011 2

1014 1

1015 2

1017 5

CUSTOMER# Ordered Books

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

1018 2

1019 1

1020 3

14 rows selected.

**Problem I**

**CODE:**

CREATE OR REPLACE VIEW BOOK\_AUTHORS AS

SELECT ISBN, LNAME, FNAME

FROM BOOKAUTHOR NATURAL JOIN AUTHOR

WITH READ ONLY;

SELECT isbn, lname, fname FROM book\_authors WHERE lname= 'WHITE';

**OUTPUT:**

View BOOK\_AUTHORS created.

ISBN LNAME FNAME

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

1915762492 WHITE WILLIAM

9247381001 WHITE WILLIAM

1915762492 WHITE LISA

2147428890 WHITE LISA

**Problem J**

**CODE:**

CREATE OR REPLACE VIEW ORDER\_SUBTOTAL AS

SELECT ORDER#, SUM(QUANTITY\*PAIDEACH) SUBTOTAL

FROM ORDERITEMS JOIN ORDERS USING (ORDER#)

GROUP BY ORDER#;

CREATE OR REPLACE VIEW ORDER\_TOTAL AS

SELECT ORDER#, SUBTOTAL, NVL(SHIPCOST, 0) SHIPPING, NVL((SUBTOTAL/RATE), 0) TAX, ROUND((SUBTOTAL + NVL(SHIPCOST, 0) + NVL((SUBTOTAL/RATE), 0)), 2) TOTAL

FROM ORDER\_SUBTOTAL JOIN ORDERS USING (ORDER#) JOIN TAX ON SHIPSTATE = STATE;

SELECT ORDER#, TOTAL

FROM ORDER\_TOTAL

WHERE TOTAL > 100;

**OUTPUT:**

View ORDER\_SUBTOTAL created.

View ORDER\_TOTAL created.

ORDER# TOTAL

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

1003 128.66

1016 101.69

1001 149.75

1011 108.81

1002 132.8

1012 200.13

1004 195.31

1007 396.59

8 rows selected.