Sam Barrionuevo ITI-2180 Foundations of Data and Database Management

**Problem A**

Input:

CREATE TABLE customers\_alt

AS

(

SELECT

customer#,

lastname,

firstname,

state,

zip "zipcode"

FROM

customers

)

Result:

Table CUSTOMERS\_ALT created.

**Problem B**

Input:

ALTER TABLE customers\_alt ADD CONSTRAINT customers\_alt\_customer#\_pk PRIMARY KEY ( customer# )

ADD CONSTRAINT tax\_state\_fk FOREIGN KEY ( state )

REFERENCES tax ( state );

Result:

Table CUSTOMERS\_ALT altered.

**Problem C**

Input:

CREATE INDEX customers\_alt\_idx ON

customers\_alt ( 'zipcode' );

SELECT

\*

FROM

"USER\_INDEXES"

WHERE

index\_name LIKE 'CUSTOMERS\_ALT\_IDX%';

DROP INDEX customers\_alt\_idx;

Result:

Index CUSTOMERS\_ALT\_IDX created.

INDEX\_NAME INDEX\_TYPE TABLE\_OWNER TABLE\_NAME TABLE\_TYPE UNIQUENES COMPRESSION PREFIX\_LENGTH TABLESPACE\_NAME INI\_TRANS MAX\_TRANS INITIAL\_EXTENT NEXT\_EXTENT MIN\_EXTENTS MAX\_EXTENTS PCT\_INCREASE PCT\_THRESHOLD INCLUDE\_COLUMN FREELISTS FREELIST\_GROUPS PCT\_FREE LOG BLEVEL LEAF\_BLOCKS DISTINCT\_KEYS AVG\_LEAF\_BLOCKS\_PER\_KEY AVG\_DATA\_BLOCKS\_PER\_KEY CLUSTERING\_FACTOR STATUS NUM\_ROWS SAMPLE\_SIZE LAST\_ANAL DEGREE INSTANCES PAR T G S BUFFER\_ FLASH\_C CELL\_FL USE DURATION PCT\_DIRECT\_ACCESS

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

ITYP\_OWNER ITYP\_NAME

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

PARAMETERS

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

GLO DOMIDX\_STATU DOMIDX FUNCIDX\_ JOI IOT DRO VISIBILIT DOMIDX\_MANAGEM SEG ORP INDEXIN AUT CON

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

CUSTOMERS\_ALT\_IDX FUNCTION-BASED NORMAL SEBARRIONUEVO CUSTOMERS\_ALT TABLE NONUNIQUE DISABLED USERS 2 255 65536 1048576 1 2147483645 10 YES 0 1 1 1 1 1 VALID 20 20 11-DEC-22 1 1 NO N N N DEFAULT DEFAULT DEFAULT NO

YES ENABLED NO NO NO VISIBLE YES NO FULL NO NO

Index CUSTOMERS\_ALT\_IDX dropped.

**Problem D**

Input:

CREATE SEQUENCE orders\_order#\_seq INCREMENT BY 2 START WITH 1022 NOCYCLE;

Result:

Sequence ORDERS\_ORDER#\_SEQ created.

Proof:

1 row inserted.

ORDER# CUSTOMER# ORDERDATE SHIPDATE SHIPSTREET SHIPCITY SH SHIPZ SHIPCOST

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

1024 11-DEC-22

**Problem E**

Input:

SELECT

title,

retail

FROM

books

WHERE

retail >= 30

AND ( pubid = 3

AND category LIKE 'COMPUTER' );

Result:

TITLE RETAIL

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

DATABASE IMPLEMENTATION 55.95

HOLY GRAIL OF ORACLE 75.95

**Problem F**

Input:

SELECT unique

firstname

|| ', '

|| lastname "CUSTOMERS IN ATLANTA, GA",

shipcity "CITY",

shipstate "STATE"

FROM

customers join orders using (customer#)

WHERE

lower (shipcity) LIKE 'atlanta%'

AND lower (shipstate) LIKE 'ga%'

Result:

CUSTOMERS IN ATLANTA, CITY ST

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

JAKE, LUCAS ATLANTA GA

**Problem G**

Input:

SELECT

title,

to\_char(retail, '$999.99') AS "EXISTING PRICE",

to\_char((round((retail \* 1.05), 0) -.01),

'$999.99') AS "PROPOSED PRICE"

FROM

books

ORDER BY

title;

Result:

TITLE EXISTING PROPOSED

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

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 EXISTING PROPOSED

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

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**

Input:

SELECT

customer#,

SUM(quantity) as "ORDERED BOOKS"

FROM

orders

JOIN orderitems USING ( order# )

GROUP BY

customer#

ORDER BY

customer#;

Result:

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**

Input:

CREATE VIEW book\_authors AS

(

SELECT

isbn,

lname,

fname

FROM

bookauthor

JOIN author USING ( authorid )

)

WITH READ ONLY;

SELECT

isbn,

lname,

fname

FROM

book\_authors

WHERE

lname LIKE 'WHITE';

Result:

View BOOK\_AUTHORS created.

ISBN LNAME FNAME

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

1915762492 WHITE WILLIAM

9247381001 WHITE WILLIAM

1915762492 WHITE LISA

2147428890 WHITE LISA

**Problem J**

Input:

create view order\_total as SELECT unique

order#,

SUM(quantity \* paideach) AS

"SUBTOTAL",

nvl(shipcost, 0) AS

"SHIPPING",

round(tax.rate, 2) AS

"TAX",

( SUM(quantity \* paideach) + nvl(shipcost, 0) ) + ( ( SUM(quantity \* paideach) + nvl(shipcost, 0) ) \* ( round(tax.rate, 2) / 100 )

) AS "TOTAL"

FROM

orders

JOIN orderitems USING ( order# )

JOIN tax ON tax.state = orders.shipstate

GROUP BY

order#,

shipcost,

rate

order by total desc;

select \* from order\_total where total > 100;

Result:

View ORDER\_TOTAL created.

ORDER# SUBTOTAL SHIPPING TAX TOTAL

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

1007 335.85 7 6.25 364.278125

1004 170.9 0 7 182.863

1012 166.4 6 6 182.744

1001 117.4 3 4 125.216

1002 111.9 3 6.25 122.08125

1003 106.85 4 6 117.501

6 rows selected.