Spool 'c:\ACC\ITSE2309\lab2.txt' append;

set echo on;

set serveroutput on;

SELECT COUNT(CUSTOMER\_NUM)

FROM CUSTOMER c

WHERE c.CUSTOMER\_NUM

NOT IN (SELECT CUSTOMER\_NUM FROM ORDERS);

Spool off;

set serveroutput off;

set echo off;

spool C:\ACC\ITSE2309\lab2.txt append

@C:\ACC\ITSE2309\lab2.sql

ITSE 2309

Distant Learning

LAB #2a and 2b

**More SQL Queries and Modification (100 points)**

Oracle 11g SQL – Chapters - 3,6,8,9,11,12,

**You will continuing using items created in Lab 1**

Lab 2a -- Problems 1 – 4,

1. For each customer that has placed an order, list each:

* The company name
* The item Description
* The manufacturer
* The quantity ordered
* The total price paid.

Include the following columns in the order given below:

* From Customer Table: Company
* From Stock Table: Description
* From the Manufact Table: Manu\_Name
* From the Items Table: Quantity, Total Price

**SQL> *SELECT C.COMPANY,***

***S.DESCRIPTION,***

***M.MANU\_NAME,***

***I.QUANTITY,***

***I.TOTAL\_PRICE***

***FROM CUSTOMER c,***

***ORDERS o,***

***ITEMS i,***

***STOCK s,***

***MANUFACT m***

***WHERE C.CUSTOMER\_NUM = O.CUSTOMER\_NUM***

***AND O.ORDER\_NUM = I.ORDER\_NUM***

***AND I.STOCK\_NUM = S.STOCK\_NUM***

***AND I.MANU\_CODE = S.MANU\_CODE***

***AND S.MANU\_CODE = M.MANU\_CODE***

***ORDER BY C.COMPANY, S.DESCRIPTION;***

COMPANY |DESCRIPTION |MANU\_NAME | QUANTITY| TOTAL\_PRICE

----------------------------|----------------------|------------------------|--------------------------|----------------------------

AA athletics |baseball gloves | Smith | 1| 450

AA athletics |volleyball | Anza | 1| 840

AA athletics |volleyball net | Anza | 5| 100

All Sports Supplies |baseball bat | Husky | 1| 240

All Sports Supplies |football | Husky | 1| 960

Gold Medal Sports |tennis ball | Smith | 1| 36

Gold Medal Sports |tennis balls | Anza | 1| 48

Kids Korner |baseball | Hero | 1| 126

Kids Korner |baseball gloves | Hero | 1| 250

Kids Korner |basketball | Hero | 1| 600

Kids Korner |football | Hero | 1| 480

Kids Korner |volleyball | Anza | 1| 840

Kids Korner |volleyball net | Anza | 10| 200

Olympic City |tennis ball | Smith | 1| 36

Olympic City |tennis balls | Anza | 1| 48

Olympic City |tennis racquet | Norge | 10| 280

Olympic City |tennis racquet | Anza | 10| 198

Play Ball! |tennis ball | Smith | 1| 36

Play Ball! |tennis balls | Anza | 1| 48

Play Ball! |tennis racquet | Anza | 5| 99

Play Ball! |tennis racquet | Anza | 1| 20

Play Ball! |tennis racquet | Anza | 5| 99

Play Ball! |volleyball | Anza | 1| 840

Play Ball! |volleyball net | Anza | 2| 40

Play Ball! |volleyball net | Anza | 1| 20

Runners and Others |tennis ball | Smith | 1| 36

Runners and Others |tennis balls | Anza | 1| 48

Runners and Others |tennis racquet | Norge | 5| 190

Runners and Others |tennis racquet | Anza | 5| 99

Runners and Others |tennis racquet |Smith | 5| 125

Sports Center |baseball gloves | Smith | 1| 450

Watson and Son |baseball | Hero | 1| 126

Watson and Son |baseball bat | Husky | 1| 240

Watson and Son |baseball gloves | Husky | 1| 800

Watson and Son |baseball gloves | Hero | 1| 960

Watson and Son |football | Hero | 1| 480

Watson and Son |football | Husky | 1| 960

38 rows selected.

Order the output by Company and Description.

Submit/hand in script and Output from SQL query

2. a) How many orders were shipped between December 25, 1999 and

January 5, 2000

a.

**SQL> *SELECT COUNT(\*) FROM ORDERS***

***WHERE ORDER\_DATE***

***BETWEEN '25/DEC/1999' AND '05/JAN/2000';***

COUNT(\*)

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

11

B) List all orders with a shipping date of January 8, 2000

Include

1) The Order Number

2) Order Date

3) Customer company name

4) Shipping Date.

Order by

Customer Company Name and Order Number.

b.

**SQL> *SELECT ORDER\_NUM,***

***ORDER\_DATE,***

***COMPANY,***

***SHIP\_DATE***

***FROM ORDERS, CUSTOMER***

***WHERE SHIP\_DATE='08-JAN-2000'***

***ORDER BY CUSTOMER.COMPANY, ORDERS.ORDER\_NUM;***

ORDER\_NUM| ORDER\_DAT |COMPANY | SHIP\_DATE

-------------- ----|--------------------|---------------------------|---------

1013| 03-JAN-00 |AA athletics |08-JAN-00

1015| 03-JAN-00 |AA athletics |08-JAN-00

1013| 03-JAN-00 |All Sports Supplies |08-JAN-00

1015| 03-JAN-00 |All Sports Supplies |08-JAN-00

1013| 03-JAN-00 |Athletic Supplies |08-JAN-00

1015| 03-JAN-00 |Athletic Supplies |08-JAN-00

1013| 03-JAN-00 |Blue Ribbon Sports |08-JAN-00

1015| 03-JAN-00 |Blue Ribbon Sports |08-JAN-00

1013| 03-JAN-00 |Gold Medal Sports |08-JAN-00

1015| 03-JAN-00 |Gold Medal Sports |08-JAN-00

1013| 03-JAN-00 |Kids Korner |08-JAN-00

1015| 03-JAN-00 |Kids Korner |08-JAN-00

1013| 03-JAN-00 |Los Altos Sports |08-JAN-00

1015| 03-JAN-00 |Los Altos Sports |08-JAN-00

1013| 03-JAN-00 |Olympic City |08-JAN-00

1015| 03-JAN-00 |Olympic City |08-JAN-00

1013| 03-JAN-00 |Phil's Sports |08-JAN-00

1015| 03-JAN-00 |Phil's Sports |08-JAN-00

1013| 03-JAN-00 |Play Ball! |08-JAN-00

1015| 03-JAN-00 |Play Ball! |08-JAN-00

1013| 03-JAN-00 |Quinn's Sports |08-JAN-00

1015| 03-JAN-00 |Quinn's Sports |08-JAN-00

1013| 03-JAN-00 |Runners and Others |08-JAN-00

1015| 03-JAN-00 |Runners and Others |08-JAN-00

1013| 03-JAN-00 |Sport Stuff |08-JAN-00

1015| 03-JAN-00 |Sport Stuff |08-JAN-00

1013| 03-JAN-00 |Sporting Place |08-JAN-00

1015| 03-JAN-00 |Sporting Place |08-JAN-00

1013| 03-JAN-00 |Sports Center |08-JAN-00

1015| 03-JAN-00 |Sports Center |08-JAN-00

1013| 03-JAN-00 |Sports Spots |08-JAN-00

1015| 03-JAN-00 |Sports Spots |08-JAN-00

1013| 03-JAN-00 |Sportstown |08-JAN-00

1015| 03-JAN-00 |Sportstown |08-JAN-00

1013| 03-JAN-00 |Watson and Son |08-JAN-00

1015| 03-JAN-00 |Watson and Son |08-JAN-00

36 rows selected.

Submit/hand in script and Output from SQL query

3. How many customers have not placed an order?

**SQL> *SELECT COUNT(c.CUSTOMER\_NUM)***

***FROM CUSTOMER c***

***WHERE c.CUSTOMER\_NUM***

***NOT IN (SELECT CUSTOMER\_NUM FROM ORDERS);***

COUNT(CUSTOMER\_NUM)

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

9

**Could also use:**

SELECT COUNT(c.CUSTOMER\_NUM)

FROM CUSTOMER c

WHERE NOT EXISTS

(select \* from orders o

where c.customer\_num = o.customer\_num);

Submit/hand in script and Output from SQL query

4.

List all customers–

I) Who are ordering equipment with a description that contains ‘ball’.

*SELECT distinct*

*customer.company, customer.CUSTOMER\_NUM, stock.stock\_num, stock.DESCRIPTION*

*FROM customer*

*INNER JOIN orders ON orders.CUSTOMER\_NUM = customer.CUSTOMER\_NUM*

*INNER JOIN items ON items.ORDER\_NUM = orders.ORDER\_NUM*

*INNER JOIN stock ON stock.STOCK\_NUM = items.STOCK\_NUM*

*WHERE STOCK.DESCRIPTION LIKE '%ball%';*

II) Include-

1) Customer number,

2) Stock number, and

3) Description.

***SQL> SELECT DISTINCT***

***ORDERS.CUSTOMER\_NUM, STOCK.STOCK\_NUM, STOCK.DESCRIPTION***

***FROM***

***ORDERS***

***JOIN***

***ITEMS ON ITEMS.ORDER\_NUM = ORDERS.ORDER\_NUM***

***JOIN***

***STOCK ON STOCK.STOCK\_NUM = ITEMS.STOCK\_NUM***

***WHERE***

***STOCK.DESCRIPTION LIKE '%ball%';***

*CUSTOMER\_NUM| STOCK\_NUM|DESCRIPTION*

--------------------|---------------------|---------------

116| 6|tennis ball

104| 6|tennis ball

101| 3|baseball bat

116| 6|tennis balls

117| 2|baseball

117| 8|volleyball

115| 6|tennis balls

117| 9|volleyball net

104| 6|tennis balls

112| 6|tennis balls

112| 6|tennis ball

117| 4|football

110| 8|volleyball

106| 4|football

106| 2|baseball

106| 3|baseball bat

110| 9|volleyball net

101| 4|football

104| 9|volleyball net

104| 8|volleyball

106| 1|baseball gloves

117| 1|baseball gloves

117| 7|basketball

115| 6|tennis ball

117| 3|baseball bat

111| 1|baseball gloves

110| 1|baseball gloves

27 rows selected.

Submit/hand in script and Output from SQL query

Do not repeat any rows.

LAB 2b

Problems 5, 6, 7 and 8

5. Use the following SQL CREATE commands to CREATE the following tables in your User ID:

CREATE TABLE Professor

(Prof\_ID NUMBER(3) Constraint pk\_Professor Primary Key,

Prof\_Lname VARCHAR2(15) NOT NULL,

Prof\_Hiredate DATE,

Prof\_Sal NUMBER(8,2),

Prof\_Dept CHAR(6)

);

CREATE TABLE Student

(Stu\_ID NUMBER(4) Constraint pk\_Student Primary Key,

Stu\_Lname VARCHAR2(15) NOT NULL,

Stu\_Major CHAR(6),

Stu\_CredHrs NUMBER(4),

Stu\_GradePts NUMBER(5),

Prof\_ID NUMBER(3),

CONSTRAINT fk\_Student\_Prof\_ID FOREIGN KEY(Prof\_ID)

REFERENCES Professor

);

Submit/Hand in: Print out of the Create commands, the system response and a

**DESCRIBE** of the tables created.

create table Professor

(Prof\_ID integer,

constraint pk\_Professor primary key(Prof\_ID),

Prof\_Lname varchar(15) not null,

Prof\_Hiredate date,

Prof\_Sal decimal(8,2),

Prof\_Dept varchar(6));

create table Student

(Stu\_ID integer ,

constraint pk\_Student primary key(STU\_ID),

Stu\_Lname varchar(15) not null,

Stu\_Major varchar(6),

Stu\_CredHrs integer,

Stu\_GradePts integer,

Prof\_ID interger,

constraint fk\_Student\_Prof\_ID foreign key (Prof\_ID)  references Professor(Prof\_ID));

**SQL> CREATE TABLE Professor**

**(Prof\_ID NUMBER(3) Constraint pk\_Professor Primary Key,**

**Prof\_Lname VARCHAR2(15) NOT NULL,**

**Prof\_Hiredate DATE,**

**Prof\_Sal NUMBER(8,2),**

**Prof\_Dept CHAR(6)**

**);**

**Table PROFESSOR created.**

**SQL> DESC PROFESSOR;**

**Name Null? Type**

**------------- -------- ------------**

**PROF\_ID NOT NULL NUMBER(3)**

**PROF\_LNAME NOT NULL VARCHAR2(15)**

**PROF\_HIREDATE DATE**

**PROF\_SAL NUMBER(8,2)**

**PROF\_DEPT CHAR(6)**

**SQL> CREATE TABLE Student**

**(Stu\_ID NUMBER(4) Constraint pk\_Student Primary Key,**

**Stu\_Lname VARCHAR2(15) NOT NULL,**

**Stu\_Major CHAR(6),**

**Stu\_CredHrs NUMBER(4),**

**Stu\_GradePts NUMBER(5),**

**Prof\_ID NUMBER(3),**

**CONSTRAINT fk\_Student\_Prof\_ID FOREIGN KEY(Prof\_ID)**

**REFERENCES Professor**

**);**

**Table STUDENT created.**

**SQL> DESC Student;**

**Name Null? Type**

**------------ -------- ------------**

**STU\_ID NOT NULL NUMBER(4)**

**STU\_LNAME NOT NULL VARCHAR2(15)**

**STU\_MAJOR CHAR(6)**

**STU\_CREDHRS NUMBER(4)**

**STU\_GRADEPTS NUMBER(5)**

**PROF\_ID NUMBER(3)**

6. Insert the following data into the tables created above using SQL INSERT commands.

NOTE: I shortened Prof\_ID = 1223 to prof\_ID=123 due to the table not allowing 4 numbers to be entered

🡪 **PROF\_ID NOT NULL NUMBER(3)**

**Another work around would have been to checke the statement to PROF\_ID NOT NULL NUMBER(4)**

Professor Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prof\_ID | Prof\_Lname | Prof\_Hiredate | Prof\_Sal | Prof\_Dept |
| 1223 | Hilbert | 20-MAY-1992 | 58000.00 | MATH |
| 243 | Newell | 15-JUL-1997 | 65500.00 | CMPSCI |
| 389 | Lessing | 04-APR-1988 | 40250.00 | ENG |

Student Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stu\_ID | Stu\_Lname | Stu\_Major | Stu\_CredHrs | Stu\_GradePts | Prof\_ID |
| 2001 | Parker | CMPSCI | 52 | 160 | 243 |
| 2166 | Smith | ENG | 30 | 75 | 389 |
| 3200 | Garcia | MATH | 62 | 248 | 123 |
| 4520 | Smith | CMPSCI | 45 | 157 | NULL |

INSERT INTO professor

VALUES ('123','Hilbert','20-MAY-1992',’58000.00','MATH');

INSERT INTO professor

VALUES ('243','Newell','15-JULY-1997','58000.00','MATH');

INSERT INTO professor

VALUES ('389','Lessing','4-APR-1988','40250.00','ENG');

INSERT INTO student

VALUES ('2001','Parker','CMPSCI','52','160','243');

INSERT INTO student

VALUES ('2166','Smith','ENG','30','75','389');

INSERT INTO student

VALUES ('3200','Garcia','MATH','62','248','123');

INSERT INTO student

VALUES ('4520','Smith','CMPSCI','45','157',NULL);

**SQL> INSERT INTO professor**

**2 VALUES ('123','Hilbert','20-MAY-1992','58000.00','MATH');**

**1 row inserted.**

**SQL> INSERT INTO professor**

**2 VALUES ('243','Newell','15-JULY-1997','58000.00','MATH');**

**1 row inserted.**

**SQL>**

**SQL> INSERT INTO professor**

**2 VALUES ('389','Lessing','4-APR-1988','40250.00','ENG');**

**1 row inserted.**

**SQL> INSERT INTO student**

**2 VALUES ('2001','Parker','CMPSCI','52','160','243');**

**1 row inserted.**

**SQL>**

**SQL> INSERT INTO student**

**2 VALUES ('2166','Smith','ENG','30','75','389');**

**1 row inserted.**

**SQL> INSERT INTO student**

**2 VALUES ('3200','Garcia','MATH','62','248','123');**

**1 row inserted.**

**SQL> INSERT INTO student**

**2 VALUES ('4520','Smith','CMPSCI','45','157',NULL);**

**1 row inserted.**

**SQL> COMMIT TRANSACTION;**

**Commit complete.**

BE SURE TO ISSUE A COMMIT AFTER TABLE MODIFICATION COMMANDS HAVE BEEN RUN SUCCESSFULLY.

Submit a

Listing of each INSERT command,

The systems response and the resulting tables after the INSERTS are completed

(Example: SELECT \* FROM Student;).

**SQL> SELECT \* FROM Professor;**

**PROF\_ID|PROF\_LNAME |PROF\_HIRE| PROF\_SAL|PROF\_D**

**----------------------|----------------------|----------------|--------------------|------------**

**123|Hilbert | 20-MAY-92| 58000|MATH**

**243|Newell | 15-JUL-97| 58000|MATH**

**389|Lessing | 04-APR-88| 40250|ENG**

**SQL> SELECT \* FROM Student;**

**STU\_ID|STU\_LNAME |STU\_MA| STU\_CREDHRS| STU\_GRADEPTS| PROF\_ID**

**---------------------|---------------------|-----------|---------------------------|----------------------------|---------------------**

**2001|Parker |CMPSCI | 52| 160| 243**

**2166|Smith |ENG | 30| 75| 389**

**4520|Smith |CMPSCI | 45| 157|**

**3200|Garcia |MATH | 62| 248| 123**

7. Perform the following SQL DELETE statements.

**Be sure to do them in order.**

Issue a COMMIT command after all DELETEs have run.

1. Try to delete Professor 389. What message do you get?

***DELETE FROM Professor WHERE PROF\_ID = 389;***

**SQL> DELETE FROM Professor WHERE PROF\_ID = 389;**

**Error starting at line : 5 File @ C:\ACC\ITSE2309\lab2.sql**

**In command -**

**DELETE FROM Professor WHERE PROF\_ID = 389**

**Error report -**

**ORA-02292: integrity constraint (ITSE2309PETRAUNGLAUB.FK\_STUDENT\_PROF\_ID) violated - child record found**

1. Delete Student 2166.

***DELETE FROM Student WHERE STU\_ID = 2166;***

***DELETE FROM Professor WHERE PROF\_ID = 389;***

1. Now Delete Professor 389. Explain why the first attempt in a. was unsuccessful, and this time the DELETE was successful.

Submit/hand in:

* A listing of the DELETE statements

**SQL> DELETE FROM Student WHERE STU\_ID = 2166;**

**1 row deleted.**

**SQL> DELETE FROM Professor WHERE PROF\_ID = 389;**

**1 row deleted.**

**SQL> commit TRANSACTION;**

**Commit complete.**

* The answers to questions b. and c.

**The delete violated the parent child / referential integrity rules. In the first delete we attempted to delete a record (Prof\_ID) from a parent table (Professor) (which is referenced by a foreign key), but a record in the child table still existed. After deleting the dependency in the Student table, we are now able to delete the entry in the Professor table.**

* A listing of the two tables after the deletes have run.

**SQL> select \* FROM Professor;**

**PROF\_ID|PROF\_LNAME |PROF\_HIRE| PROF\_SAL|PROF\_D**

**-------------------------|-------------------------|-----------------|--------------------------|-------------------**

**123|Hilbert |20-MAY-92 | 58000|MATH**

**243|Newell |15-JUL-97 | 58000|MATH**

**SQL> select \* from Student;**

**STU\_ID|STU\_LNAME |STU\_MA | STU\_CREDHRS| STU\_GRADEPTS| PROF\_ID**

**--------------|------------------------|-------------|-------------------------------|--------------------------------|--------------------**

**2001|Parker |CMPSCI | 52| 160| 243**

**4520|Smith |CMPSCI | 45| 157|**

**3200|Garcia |MATH | 62| 248| 123**

8. Perform the following UPDATE commands.

Issue a COMMIT command after each UPDATEs have run.

1. Make the Prof\_ID for Student 4520 be 1223.

**SQL> UPDATE Student**

**SET PROF\_ID = 1223**

**WHERE STUD\_ID = 5420;**

**Error starting at line : 1 in command -**

**update STUDENT**

**set PROF\_ID = 1223**

**where STU\_ID = 4520**

**Error report -**

**ORA-01438: value larger than specified precision allowed for this column**

The error is due to the number 1223 having more characters than are allowed for the column;

SQL>DESC STUDENT;

Name Null? Type

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

STU\_ID NOT NULL NUMBER(4)

STU\_LNAME NOT NULL VARCHAR2(15)

STU\_MAJOR CHAR(6)

STU\_CREDHRS NUMBER(4)

STU\_GRADEPTS NUMBER(5)

**PROF\_ID NUMBER(3)**

**SQL> select \* from STUDENT;**

**STU\_ID STU\_LNAME STU\_MA STU\_CREDHRS STU\_GRADEPTS PROF\_ID**

**------------- --------------------- ------------ ------------------- ----------------------- ----------------**

**2001 Parker CMPSCI 52 160 243**

**4520 Smith CMPSCI 45 157**

**3200 Garcia MATH 62 248 123**

1. Change what each professor is paid to be two and a half times what they earn now

Submit/hand in: A listing of the UPDATE statements

A listing of the two tables after the UPDATEs have run.

**update PROFESSOR**

**set PROF\_SAL = PROF\_SAL\*2.5;**

**SQL> update PROFESSOR**

**set PROF\_SAL = PROF\_SAL\*2.5;**

**2 rows updated.**

**SQL> commit TRANSACTION;**

**Commit complete.**

**SQL> select \* from Professor;**

**PROF\_ID PROF\_LNAME PROF\_HIRE PROF\_SAL PROF\_D**

**-------------- ----------------------- --------------- ----------------- ---------------**

**123 Hilbert 20-MAY-92 145000 MATH**

**243 Newell 15-JUL-97 145000 MATH**