##########################3-1A########################

SQL> CREATE TABLE Stu

2 (sname CHAR(5),

3 majorCode VARCHAR2(2));

Table created.

SQL> CREATE TABLE Major

2 (majorCode VARCHAR2(2),

3 majorDesc CHAR(16));

Table created.

SQL> INSERT INTO Stu

2 VALUES ('Jones', 'CS');

1 row created.

SQL> INSERT INTO Stu

2 VALUES ('Smith', 'AC');

1 row created.

SQL> INSERT INTO Stu

2 VALUES ('Evans', 'MA');

1 row created.

SQL> INSERT INTO Stu

2 VALUES ('Adams', 'CS');

1 row created.

SQL> SELECT \*

2 FROM Stu;

SNAME MAJORC

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

Jones CS

Smith AC

Evans MA

Adams CS

SQL> INSERT INTO Major

2 VALUES ('AC', 'Accounting');

1 row created.

SQL> INSERT INTO Major

2 VALUES ('CS', 'Computer Science');

1 row created.

SQL> INSERT INTO Major

2 VALUES ('MA', 'Math');

1 row created.

MAJORC MAJORDESC

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

AC Accounting

CS Computer Science

MA Math

SQL> SELECT \*

2 FROM Stu, Major;

SNAME MAJORC MAJORC MAJORDESC

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

Jones CS AC Accounting

Smith AC AC Accounting

Evans MA AC Accounting

Adams CS AC Accounting

Jones CS CS Computer Science

Smith AC CS Computer Science

Evans MA CS Computer Science

Adams CS CS Computer Science

Jones CS MA Math

Smith AC MA Math

Evans MA MA Math

SNAME MAJORC MAJORC MAJORDESC

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

Adams CS MA Math

12 rows selected.

\*\*\*\*\*\*\*\* I got 12 rows because the result will always will always be n\*m rows.

########################3-1B##################

SQL> SELECT \*

2 FROM Stu s, Major m

3 WHERE s.majorCode=m.majorCode;

SNAME MAJORC MAJORC MAJORDESC

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

Jones CS CS Computer Science

Smith AC AC Accounting

Evans MA MA Math

Adams CS CS Computer Science

SQL> SELECT \*

2 FROM Stu s INNER JOIN Major m

3 ON s.majorCode=m.majorCode;

SNAME MAJORC MAJORC MAJORDESC

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

Jones CS CS Computer Science

Smith AC AC Accounting

Evans MA MA Math

Adams CS CS Computer Science

\*\*\*\*\*\*\*\*\*I got 4 rows.

##############################3-1C#########################

SQL> SELECT \*

2 FROM Stu s, Major m

3 WHERE majorCode=majorCode;

WHERE majorCode=majorCode

\*

ERROR at line 3:

ORA-00918: column ambiguously defined

\*\*\*\*\*\*\*The columns are named the same in both tables, so the system is confused.

##################3-1D#####################

SQL> SELECT COUNT (\*)

2 FROM Stu s, Major m

3 WHERE s.majorCode=m.majorCode;

COUNT(\*)

----------

4

SQL> SELECT COUNT (\*)

2 FROM Stu, Major

3 --WHERE Stu.majorCode=Major.majorCode;

COUNT(\*)

----------

12

#########################3-1E####################

SQL> INSERT INTO Major

2 VALUES ('IT', 'Information Tech');

1 row created.

SQL> INSERT INTO Major

2 VALUES ('ST', 'Statistics');

1 row created.

SQL> SELECT \*

2 FROM Stu RIGHT OUTER JOIN Major

3 ON Stu.majorCode=Major.majorCode;

SNAME MAJORC MAJORC MAJORDESC

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

Jones CS CS Computer Science

Smith AC AC Accounting

Evans MA MA Math

Adams CS CS Computer Science

IT Information Tech

ST Statistics

6 rows selected.

SQL> SELECT \*

2 FROM Stu, Major

3 WHERE Stu.majorCode(+)=Major.majorCode;

SNAME MAJORC MAJORC MAJORDESC

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

Jones CS CS Computer Science

Smith AC AC Accounting

Evans MA MA Math

Adams CS CS Computer Science

IT Information Tech

ST Statistics

6 rows selected.

###############################3-1F######################

SQL> INSERT INTO Stu (sname)

2 VALUES ('Arpan');

1 row created.

SQL> INSERT INTO Stu (sname)

2 VALUES ('Ayona');

1 row created.

SQL> SELECT \*

2 FROM Stu LEFT OUTER JOIN Major

3 ON Stu.majorCode=Major.majorCode;

SNAME MAJORC MAJORC MAJORDESC

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

Smith AC AC Accounting

Adams CS CS Computer Science

Jones CS CS Computer Science

Evans MA MA Math

Ayona

Arpan

6 rows selected.

SQL> SELECT \*

2 FROM Stu, Major

3 WHERE Stu.majorCode=Major.majorCode(+);

SNAME MAJORC MAJORC MAJORDESC

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

Smith AC AC Accounting

Adams CS CS Computer Science

Jones CS CS Computer Science

Evans MA MA Math

Ayona

Arpan

6 rows selected.

###############################3-1G##################

SQL> SELECT \*

2 FROM Stu FULL OUTER JOIN Major

3 ON Stu.majorCode=Major.majorCode;

SNAME MAJORC MAJORC MAJORDESC

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

Jones CS CS Computer Science

Smith AC AC Accounting

Evans MA MA Math

Adams CS CS Computer Science

Arpan

Ayona

IT Information Tech

ST Statistics

8 rows selected.

#############################3-2A###################

SQL> CREATE TABLE T1

2 (ename CHAR(16),

3 jobno NUMBER(3));

Table created.

SQL> CREATE TABLE T2

2 (jobno NUMBER(3),

3 jobdesc CHAR(16));

Table created.

SQL> INSERT INTO T1 (jobno)

2 VALUES (100);

1 row created.

SQL> INSERT INTO T1 (jobno)

2 VALUES (200);

1 row created.

SQL> INSERT INTO T1 (jobno)

2 VALUES (300);

1 row created.

SQL> INSERT INTO T2 (jobno)

2 VALUES (100);

1 row created.

SQL> INSERT INTO T2 (jobno)

2 VALUES (200);

1 row created.

SQL> SELECT \*

2 FROM T1, T2

3 WHERE T1.jobno=T2.jobno;

ENAME JOBNO JOBNO JOBDESC

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

100 100

200 200

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2 ROWS ON EQUI-JOIN

###############################3-2B#####################

\*\*\*\*\*\*\*\*\*\*\*\*\*I would only expect 1 row because even though the tables would match on two rows of T2, both rows are identical. Therefore, one would be thrown out.

##############################3-2C######################

\*\*\*\*\*\*\*\*\*\*\*\*No rows would match (Zero Rows)

##########################3-2D#####################

\*\*\*\*\*\*\*\*\*\*\*\*\* 1 Row

###########################3-2E###################

An equi-join would produce the number of rows where the data matches

A Cartesian product would produce the product of multiplying the number of rows from the first table with the number of rows from the second (n\*m)

###########################3-2F###################

0-(N or M whichever is larger)

This depends on how many rows have matching data points.

#########################3-3A#####################

SQL> CREATE TABLE T3

2 (jobdesc CHAR(16),

3 minpay NUMBER(6,2));

Table created.

SQL> UPDATE T2

2 SET jobdesc='Driver'

3 WHERE jobno=100;

1 row updated.

SQL> UPDATE T2

2 SET jobdesc='Loader'

3 WHERE jobno=200;

1 row updated.

SQL> INSERT INTO T3 (jobdesc)

2 VALUES ('Driver');

1 row created.

SQL> INSERT INTO T3 (jobdesc)

2 VALUES ('Loader');

1 row created.

SQL> INSERT INTO T3 (jobdesc)

2 VALUES ('Manager');

1 row created.

SQL> SELECT \*

2 FROM T1, T2, T3

3 WHERE T1.jobno=T2.jobno

4 AND T2.jobdesc=T3.jobdesc;

ENAME JOBNO JOBNO JOBDESC

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

JOBDESC MINPAY

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

100 100 Driver

Driver

200 200 Loader

Loader

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2 ROWS \*\*\*\*\*\*\*\*

##############################3-3B ###################

SQL> SELECT COUNT (\*)

2 FROM T1, T2, T3

3 WHERE T1.jobno=T2.jobno

4 AND T2.jobdesc=T3.jobdesc;

COUNT(\*)

----------

2

##############################3-3C#################

SQL> SELECT COUNT (\*)

2 FROM T1, T2, T3;

COUNT(\*)

----------

18

############################3-3D#####################

N Minus one

##########################3-4A###################

SQL> SELECT COUNT (\*)

2 FROM DICT;

COUNT(\*)

----------

835

#########################3-4B##################

SQL> SELECT COUNT (\*)

2 FROM USER\_TABLES;

COUNT(\*)

----------

6

########################3-4C####################

SQL> SELECT COUNT (\*)

2 FROM ALL\_SYNONYMS;

COUNT(\*)

----------

30252

#######################3-4D#################

SQL> DESC USER\_TABLES

Name Null? Type

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

TABLE\_NAME NOT NULL VARCHAR2(30)

TABLESPACE\_NAME VARCHAR2(30)

CLUSTER\_NAME VARCHAR2(30)

IOT\_NAME VARCHAR2(30)

STATUS VARCHAR2(8)

PCT\_FREE NUMBER

PCT\_USED NUMBER

INI\_TRANS

SQL> DESC USER\_OBJECTS

Name Null? Type

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

OBJECT\_NAME VARCHAR2(128)

SUBOBJECT\_NAME VARCHAR2(30)

OBJECT\_ID NUMBER

DATA\_OBJECT\_ID NUMBER

OBJECT\_TYPE VARCHAR2(19)

SQL> DESC USER\_VIEWS

Name Null? Type

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

VIEW\_NAME NOT NULL VARCHAR2(30)

TEXT\_LENGTH NUMBER

TEXT LONG

TYPE\_TEXT\_LENGTH NUMBER

TYPE\_TEXT VARCHAR2(4000)

OID\_TEXT\_LENGTH NUMBER

SQL> DESC USER\_SYNONYMS

Name Null? Type

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

SYNONYM\_NAME NOT NULL VARCHAR2(30)

TABLE\_OWNER VARCHAR2(30)

TABLE\_NAME NOT NULL VARCHAR2(30)

DB\_LINK VARCHAR2(128)

SQL> DESC ALL\_TABLES

Name Null? Type

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

OWNER NOT NULL VARCHAR2(30)

TABLE\_NAME NOT NULL VARCHAR2(30)

TABLESPACE\_NAME VARCHAR2(30)

CLUSTER\_NAME VARCHAR2(30)

IOT\_NAME VARCHAR2(30)

STATUS VARCHAR2(8)

PCT\_FREE NUMBER

\*\*\*\*\*\*\*\*\*\*\*\*\*\*In ALL\_TABLES there is an Owner attribute

#################################3-5#####################

SQL> SELECT s.sname "Student Name", g.grade "Grade Assigned"

2 FROM Stu s, Grrep g

3 WHERE s.stno=g.student\_number

4 AND rownum < 11;

Student Name Grade Assigned

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

Lineas D

Lineas B

Lineas B

Lineas A

Lineas B

Mary A

Mary B

Mary B

Mary B

Mary B

10 rows selected.

###########################3-6###################

SQL> drop table stu;

Table dropped.

SQL> DROP TABLE T1;

Table dropped.

SQL> DROP TABLE T2;

Table dropped.

SQL> DROP TABLE T3;

Table dropped.

SQL> DROP TABLE Major;

Table dropped.

##################################3-7##########################

SQL> SELECT sname

2 FROM Stu

3 WHERE class>3;

SNAME

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

Mary

Kelly

Donald

Chris

Holly

Jerry

Harrison

Francis

Jake

Benny

10 rows selected.