Name: _____Skylar Trendley___________________CS 5300 Chapter 6 and 7 Homework

1) Complete the table below with the Data Type you would use for the listed attribute (5 pts).

Attribute	**Data Types**
Social Security Number.	CHAR(9)
A field used to flag if a package has been picked up or not. Assume only two options.	BOOLEAN
The time field on a monthly police call log.	TIMESTAMP
Student Identification number.	CHAR(8)
Hourly temperature with up to 2 decimal places.	NUMERIC(DOUBLE)
Newspaper Names	VARCHAR(20)

Use the following tables, ART and ORIGIN, to answer all parts of question 2 and 3.

ART

ITEM	**NAME**	**ARTIST**	**ORIGIN_ID**	**DATING**	**MEDIA**
9182	One Hundred Horses	Lang Shining	1111	960 to 1127	Painting
6922	The Great Wave off Kanagawa	Katsushika Hokusai	6943	1829 to 1833	Painting
2049	Toluvila statue	NULL	8415	300 to 400	Statuary
2038	Sasanian silver vessel	NULL	1598	700 to 722	Silver
3964	Nymph of the Luo River	Gu Kaizhi	1111	317 to 420	Painting
3097	The Hunt of the Unicorn	NULL	3543	1680	Tapestries

ORIGIN

ORIGIN_ID	**LOCATION**
1111	China
3543	France
6943	Japan
8415	Sri Lanka
1598	Tajikistan

2) Use the above tables ART and ORIGIN to answer parts a, b, c, and d.

a) Complete the table below with the Data Type you would use for the listed columns (7 pts).

Attribute	**Data Type**
ART.ITEM	CHAR(4)
ART.NAME	VARCHAR(25)
ART.ARTIST	VARCHAR(25)
ART.ORIGIN_ID	CHAR(4)
ART.DATING	For this I would WANT to use a DATE data type but by the way its formatted with the word “to” in it, I would likely use CHAR(25)
ART.MEDIA	VARCHAR(15)
ORIGIN.ORIGIN_ID	CHAR(4)
ORIGIN.LOCATION	VARCHAR(25)

b) Assume the development policies of the corporation you are working for require you to add Primary Keys in an ALTER TABLE commands separate from the CREATE TABLE. Write the statements that you would use to create ART and ORIGIN in a database and add the Primary Keys (10 pts).

```
DROP TABLE ART;
DROP TABLE ORIGIN;

CREATE TABLE ORIGIN
(
        Origin_ID       CHAR(4)             NOT NULL,
        Location        VARCHAR(25)         NOT NULL
);

ALTER TABLE ORIGIN ADD PRIMARY KEY (Origin_ID);

CREATE TABLE ART
(
        Item            CHAR(4)             NOT NULL,
        Origin_ID       CHAR(4)             NOT NULL,
        Name            VARCHAR(35),
        Artist          VARCHAR(35),
        Dating          VARCHAR(35),
        Media           VARCHAR(35)
);
ALTER TABLE ART ADD PRIMARY KEY (Item);
```

c) The ORIGIN_ID in the ART table is a Foreign Key the reference the ORIGIN_ID in the ORIGIN table. Write the ALTER TABLE statement you would use to define the Foreign Key constraint (5 pts).

```
ALTER TABLE ART
        ADD CONSTRAINT fk_origin
                FOREIGN KEY(Origin_ID)
                REFERENCES ORIGIN(Origin_ID)
                ON DELETE CASCADE;
```

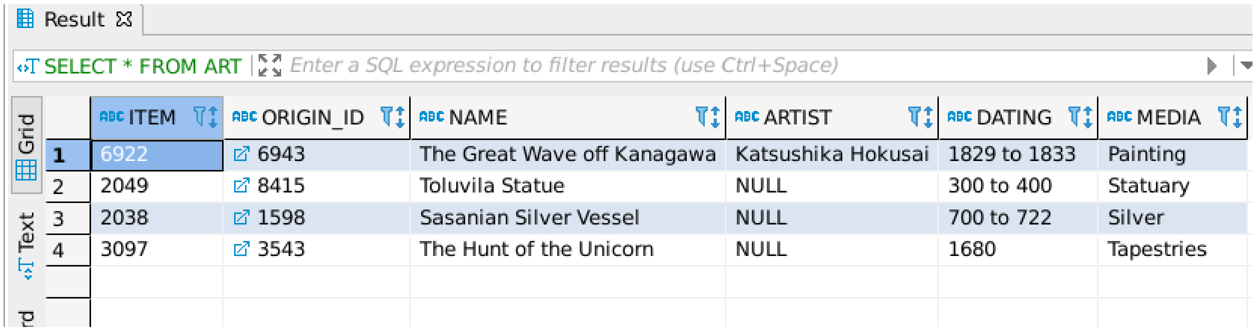
d) Execute your CREATE TABLE statements and ALTER TABLE statements from part b and c on the SCSP Oracle database and grant KOOBP access to the tables with the following two statements (4 pts).

```
        GRANT SELECT ON ART TO KOOBP;
        GRANT SELECT ON ORIGIN TO KOOBP;
```

3) Use the above tables ART and ORIGIN to answer the following parts:

a) The traveling “Art of China” exhibit at the very small museum using the database detailed above has moved to another museum. So, all artwork from China is no longer at the museum. Write the DELETE statement you would use to remove China from the ORIGIN table (5 pts).

```
DELETE
FROM
        ORIGIN
WHERE
        Location = ‘China’;
```



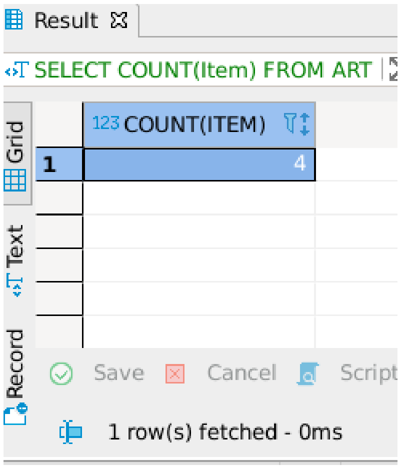
SELECT * FROM ART

Enter a SQL expression to filter results (use Ctrl+Space)

	ITEM	ORIGIN_ID	NAME	ARTIST	DATING	MEDIA
1	6922	6943	The Great Wave off Kanagawa	Katsushika Hokusai	1829 to 1833	Painting
2	2049	8415	Toluvila Statue	NULL	300 to 400	Statuary
3	2038	1598	Sasanian Silver Vessel	NULL	700 to 722	Silver
4	3097	3543	The Hunt of the Unicorn	NULL	1680	Tapestries

b) Assuming the Foreign Key constraint from part c had ON DELETE CASCADE, if you were to execute the DELETE statement from part e, what would be the results of the following statement (3 pts):

```
SELECT COUNT(ITEM)
FROM ART;
```



SELECT COUNT(Item) FROM ART

	COUNT(ITEM)
1	4

1 row(s) fetched - 0ms

The output is 4.

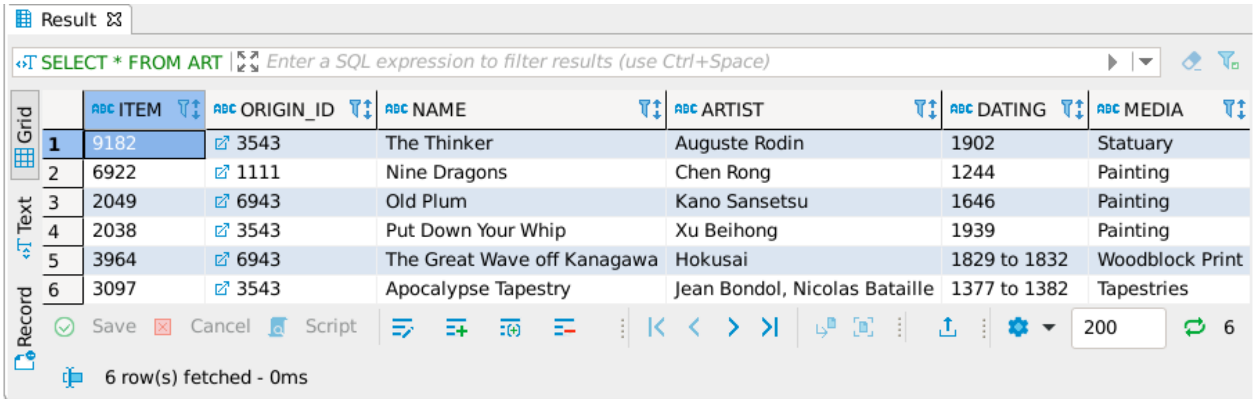
4) Use the above tables ART and ORIGIN to answer the following:

a) Write the transaction to DELETE all items from the ART table and INSERT 5 of your favorite Art pieces into the database. Show the statements use and execute them on SCSP (10 pts).

```
DELETE
FROM ART;

INSERT ALL
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media)
VALUES('9182','The Thinker','Auguste Rodin','3543','1902','Statuary')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media)
VALUES('6922','Nine Dragons','Chen Rong','1111','1244','Painting')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media)
VALUES('2049','Old Plum','Kano Sansetsu','6943','1646','Painting')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media)
VALUES('2038','Put Down Your Whip','Xu Beihong','3543','1939','Painting')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media)
VALUES('3964','The Great Wave off Kanagawa','Hokusai','6943','1829 to
1832','Woodblock Print')
```

```
            INTO ART (Item,Name,Artist,Origin_ID,Dating,Media)
    VALUES('3097','Apocalypse Tapestry','Jean Bondol, Nicolas
    Bataille','3543','1377 to 1382','Tapestries')
            SELECT * FROM DUAL;
```



SELECT * FROM ART | Enter a SQL expression to filter results (use Ctrl+Space)

	ITEM	ORIGIN_ID	NAME	ARTIST	DATING	MEDIA
1	9182	3543	The Thinker	Auguste Rodin	1902	Statuary
2	6922	1111	Nine Dragons	Chen Rong	1244	Painting
3	2049	6943	Old Plum	Kano Sansetsu	1646	Painting
4	2038	3543	Put Down Your Whip	Xu Beihong	1939	Painting
5	3964	6943	The Great Wave off Kanagawa	Hokusai	1829 to 1832	Woodblock Print
6	3097	3543	Apocalypse Tapestry	Jean Bondol, Nicolas Bataille	1377 to 1382	Tapestries

6 row(s) fetched - 0ms

5) Use the above tables ART and ORIGIN to answer this question:

You are asked to convert the LOCATION column in the ORIGIN table to two separate columns, COUNTRY and CITY. Write the statements you would need to:

- Add two new columns COUNTRY and CITY to the ORIGIN table,
- Copy the data in the LOCATION column to the COUNTRY column,
- Delete the LOCATION column.

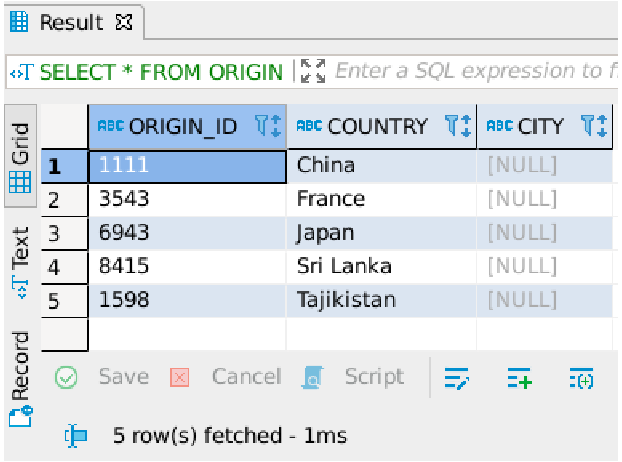
Show the statements used and then execute them on SCSP (12 pts).

```
ALTER TABLE ORIGIN
        ADD
        (
                COUNTRY VARCHAR(35),
                CITY VARCHAR(35)
        );


UPDATE ORIGIN
        SET COUNTRY=LOCATION;

ALTER TABLE ORIGIN
        DROP COLUMN LOCATION;

SELECT * FROM ORIGIN;
```



SELECT * FROM ORIGIN

	ORIGIN_ID	COUNTRY	CITY
1	1111	China	[NULL]
2	3543	France	[NULL]
3	6943	Japan	[NULL]
4	8415	Sri Lanka	[NULL]
5	1598	Tajikistan	[NULL]

5 row(s) fetched - 1ms

6. Connect to the scsp database and use the os_hr tables to answer the following questions.

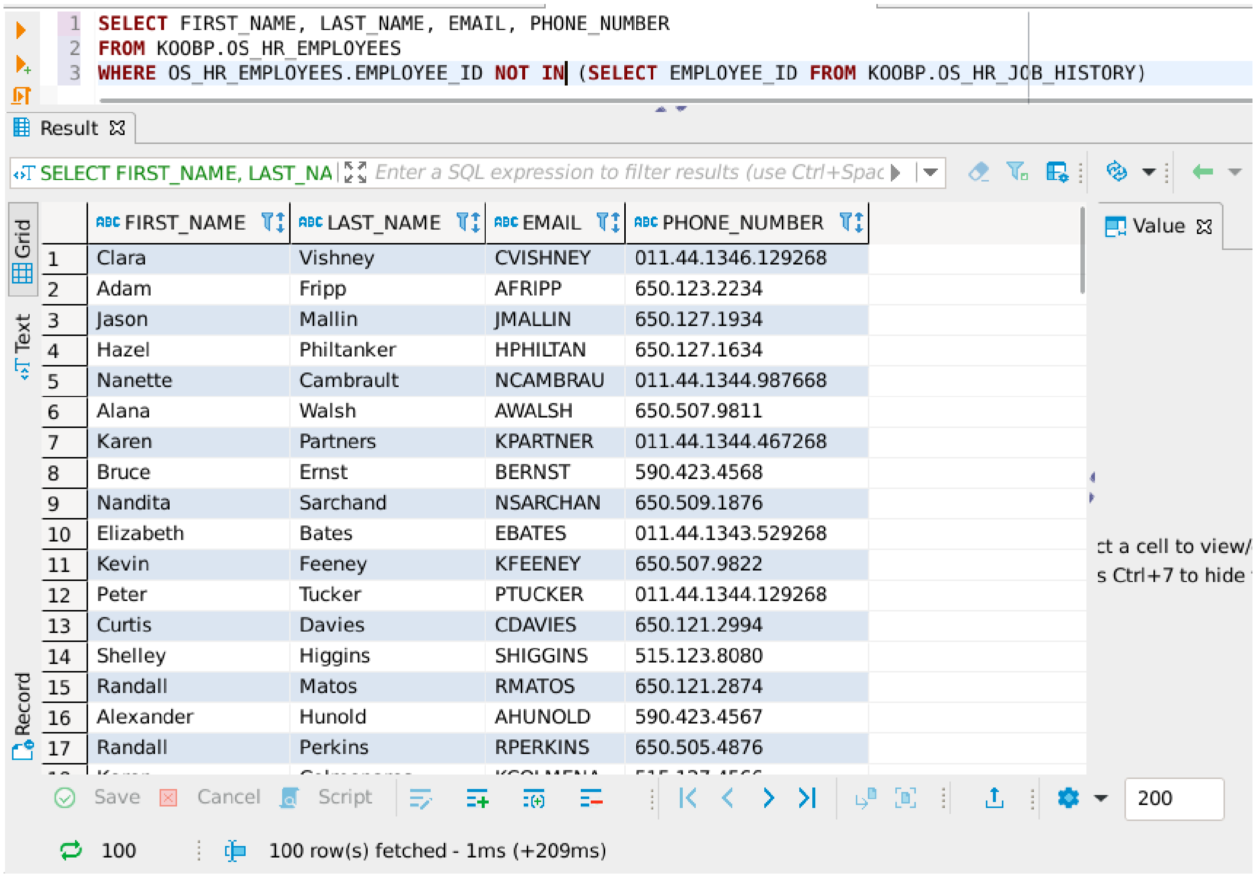
   KOOBP.OS_HR_JOBS
   KOOBP.OS_HR_JOB_HISTORY
   KOOBP.OS_HR_DEPARTMENTS
   KOOBP.OS_HR_EMPLOYEES
   KOOBP.OS_HR_LOCATIONS
   KOOBP.OS_HR_REGIONS
   KOOBP.OS_HR_COUNTRIES

   For each answer, show the SQL statement you used to get it and a screen shot of the results box.

   a. New information has been found pertaining to the employee contact list, from the previous homework, and the query we used for the employee contact list is wrong.

      The contact list has people that no longer work for the company. For example Lex De Haan in on the contact list, but no longer with the company. The KOOBP.OS_HR_JOB_HISTORY table has a list of employees who are no longer working from the company. Using a nested query, get a corrected employee contact list with first name, last name, email, and phone number for each currently employed person. Show the statement used and a screen shot of the results box (3 pts).

SELECT FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER
FROM KOOBP.OS_HR_EMPLOYEES
WHERE OS_HR_EMPLOYEES.EMPLOYEE_ID NOT IN (SELECT EMPLOYEE_ID FROM KOOBP.OS_HR_JOB_HISTORY)



```
SELECT FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER
FROM KOOBP.OS_HR_EMPLOYEES
WHERE OS_HR_EMPLOYEES.EMPLOYEE_ID NOT IN (SELECT EMPLOYEE_ID FROM KOOBP.OS_HR_JOB_HISTORY)
```

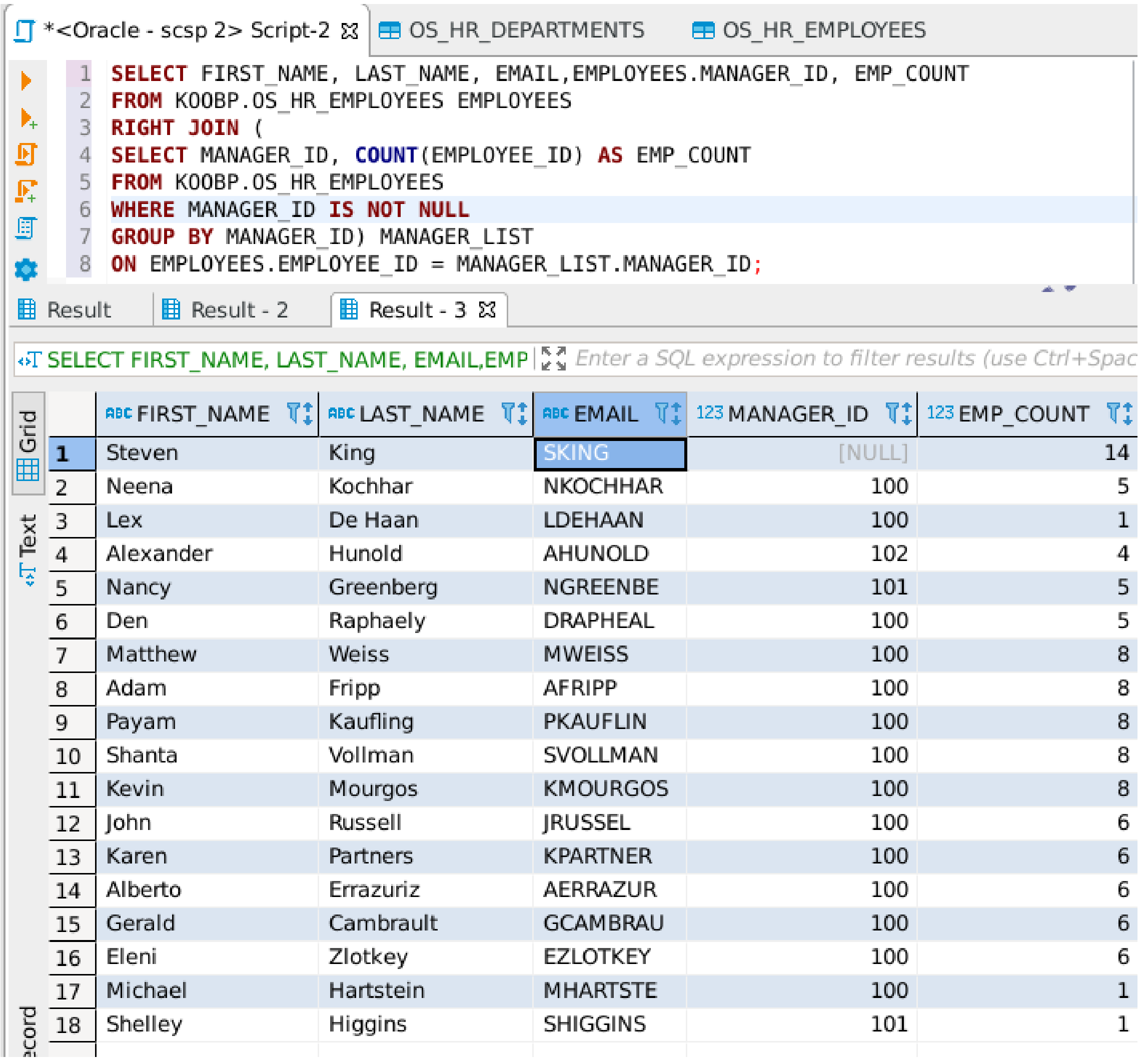
Result
SELECT FIRST_NAME, LAST_NA
Enter a SQL expression to filter results (use Ctrl+Spac
Grid
Text
Record
Value
ct a cell to view/
s Ctrl+7 to hide
Save
Cancel
Script
200

	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	Clara	Vishney	CVISHNEY	011.44.1346.129268
2	Adam	Fripp	AFRIPP	650.123.2234
3	Jason	Mallin	JMALLIN	650.127.1934
4	Hazel	Philtanker	HPHILTAN	650.127.1634
5	Nanette	Cambrault	NCAMBRAU	011.44.1344.987668
6	Alana	Walsh	AWALSH	650.507.9811
7	Karen	Partners	KPARTNER	011.44.1344.467268
8	Bruce	Ernst	BERNST	590.423.4568
9	Nandita	Sarchand	NSARCHAN	650.509.1876
10	Elizabeth	Bates	EBATES	011.44.1343.529268
11	Kevin	Feeney	KFEENEY	650.507.9822
12	Peter	Tucker	PTUCKER	011.44.1344.129268
13	Curtis	Davies	CDAVIES	650.121.2994
14	Shelley	Higgins	SHIGGINS	515.123.8080
15	Randall	Matos	RMATOS	650.121.2874
16	Alexander	Hunold	AHUNOLD	590.423.4567
17	Randall	Perkins	RPERKINS	650.505.4876
18	Karen	Colmenares	KCOLMENA	515.127.4566

100   100 row(s) fetched - 1ms (+209ms)

b. Using nested queries, a GROUP BY clause, and Aggregate functions, get a list of manager's first names, last names, email, manager_id, and the count of employees who report to them. Show the statement used and a screen capture of the results box (5 pts).

SELECT FIRST_NAME, LAST_NAME, EMAIL,EMPLOYEES.MANAGER_ID, EMP_COUNT
FROM KOOBP.OS_HR_EMPLOYEES EMPLOYEES
RIGHT JOIN (
SELECT MANAGER_ID, COUNT(EMPLOYEE_ID) AS EMP_COUNT
FROM KOOBP.OS_HR_EMPLOYEES
WHERE MANAGER_ID IS NOT NULL
GROUP BY MANAGER_ID) MANAGER_LIST
ON EMPLOYEES.EMPLOYEE_ID = MANAGER_LIST.MANAGER_ID;


*<Oracle - scsp 2> Script-2
OS_HR_DEPARTMENTS
OS_HR_EMPLOYEES
Result
Result - 2
Result - 3
Enter a SQL expression to filter results (use Ctrl+Spac
Grid
Text
ecord

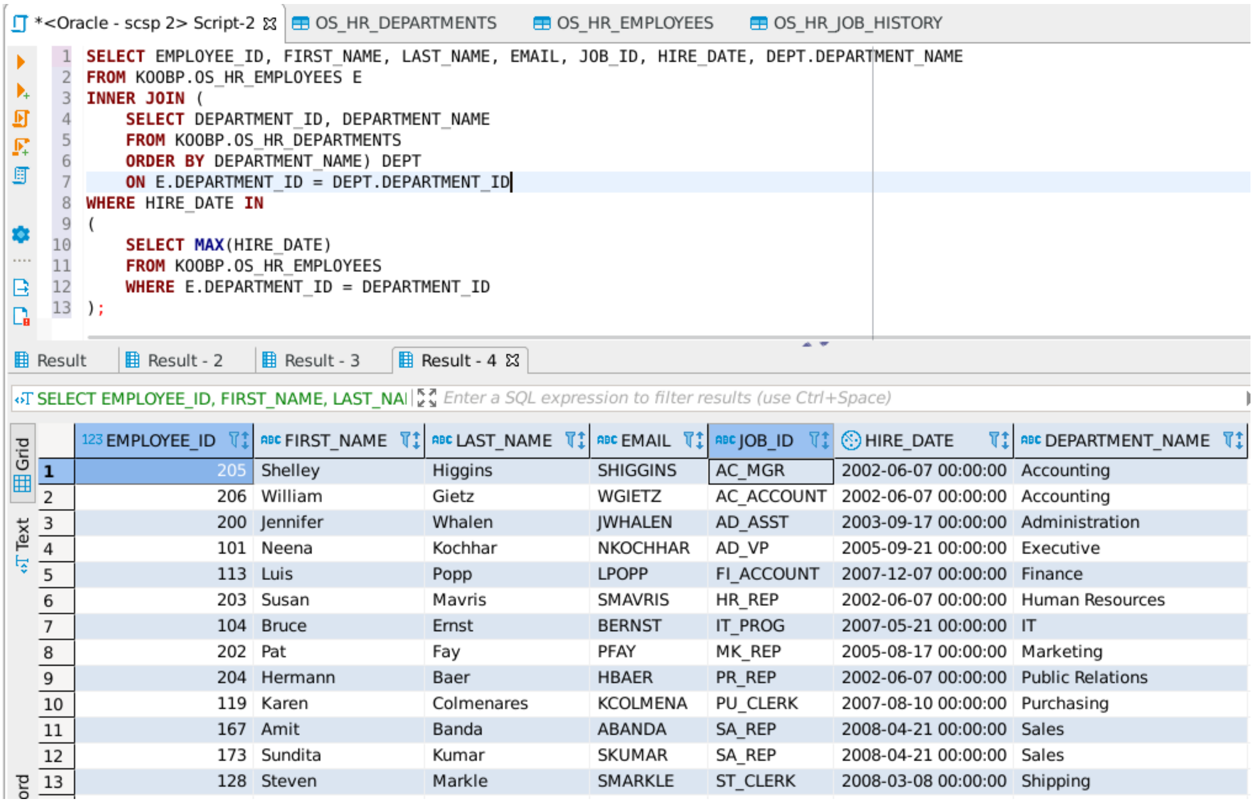
```
SELECT FIRST_NAME, LAST_NAME, EMAIL,EMPLOYEES.MANAGER_ID, EMP_COUNT
FROM KOOBP.OS_HR_EMPLOYEES EMPLOYEES
RIGHT JOIN (
SELECT MANAGER_ID, COUNT(EMPLOYEE_ID) AS EMP_COUNT
FROM KOOBP.OS_HR_EMPLOYEES
WHERE MANAGER_ID IS NOT NULL
GROUP BY MANAGER_ID) MANAGER_LIST
ON EMPLOYEES.EMPLOYEE_ID = MANAGER_LIST.MANAGER_ID;
```

SELECT FIRST_NAME, LAST_NAME, EMAIL,EMP

	FIRST_NAME	LAST_NAME	EMAIL	MANAGER_ID	EMP_COUNT
1	Steven	King	SKING	[NULL]	14
2	Neena	Kochhar	NKOCHHAR	100	5
3	Lex	De Haan	LDEHAAN	100	1
4	Alexander	Hunold	AHUNOLD	102	4
5	Nancy	Greenberg	NGREENBE	101	5
6	Den	Raphaely	DRAPHEAL	100	5
7	Matthew	Weiss	MWEISS	100	8
8	Adam	Fripp	AFRIPP	100	8
9	Payam	Kaufling	PKAUFLIN	100	8
10	Shanta	Vollman	SVOLLMAN	100	8
11	Kevin	Mourgos	KMOURGOS	100	8
12	John	Russell	JRUSSEL	100	6
13	Karen	Partners	KPARTNER	100	6
14	Alberto	Errazuriz	AERRAZUR	100	6
15	Gerald	Cambrault	GCAMBRAU	100	6
16	Eleni	Zlotkey	EZLOTKEY	100	6
17	Michael	Hartstein	MHARTSTE	100	1
18	Shelley	Higgins	SHIGGINS	101	1

c. Using correlated nested queries, show the employee_id, first name, last name, email, job id, department name, and hire date for the most recent hires for each department, ordered by department name. Show the statement and a screen capture of the results box (8 pts).

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, JOB_ID, HIRE_DATE,
DEPT.DEPARTMENT_NAME
FROM KOOBP.OS_HR_EMPLOYEES E
INNER JOIN (
        SELECT DEPARTMENT_ID, DEPARTMENT_NAME
        FROM KOOBP.OS_HR_DEPARTMENTS
        ORDER BY DEPARTMENT_NAME) DEPT
        ON E.DEPARTMENT_ID = DEPT.DEPARTMENT_ID
WHERE HIRE_DATE IN
(
        SELECT MAX(HIRE_DATE)
        FROM KOOBP.OS_HR_EMPLOYEES
        WHERE E.DEPARTMENT_ID = DEPARTMENT_ID
);
```



```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, JOB_ID, HIRE_DATE, DEPT.DEPARTMENT_NAME
FROM KOOBP.OS_HR_EMPLOYEES E
INNER JOIN (
    SELECT DEPARTMENT_ID, DEPARTMENT_NAME
    FROM KOOBP.OS_HR_DEPARTMENTS
    ORDER BY DEPARTMENT_NAME) DEPT
    ON E.DEPARTMENT_ID = DEPT.DEPARTMENT_ID
WHERE HIRE_DATE IN
(
    SELECT MAX(HIRE_DATE)
    FROM KOOBP.OS_HR_EMPLOYEES
    WHERE E.DEPARTMENT_ID = DEPARTMENT_ID
);
```

SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, JOB_ID, HIRE_DATE, DEPT.DEPARTMENT_NAME...

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	JOB_ID	HIRE_DATE	DEPARTMENT_NAME
1	205	Shelley	Higgins	SHIGGINS	AC_MGR	2002-06-07 00:00:00	Accounting
2	206	William	Gietz	WGIETZ	AC_ACCOUNT	2002-06-07 00:00:00	Accounting
3	200	Jennifer	Whalen	JWHALEN	AD_ASST	2003-09-17 00:00:00	Administration
4	101	Neena	Kochhar	NKOCHHAR	AD_VP	2005-09-21 00:00:00	Executive
5	113	Luis	Popp	LPOPP	FI_ACCOUNT	2007-12-07 00:00:00	Finance
6	203	Susan	Mavris	SMAVRIS	HR_REP	2002-06-07 00:00:00	Human Resources
7	104	Bruce	Ernst	BERNST	IT_PROG	2007-05-21 00:00:00	IT
8	202	Pat	Fay	PFAY	MK_REP	2005-08-17 00:00:00	Marketing
9	204	Hermann	Baer	HBAER	PR_REP	2002-06-07 00:00:00	Public Relations
10	119	Karen	Colmenares	KCOLMENA	PU_CLERK	2007-08-10 00:00:00	Purchasing
11	167	Amit	Banda	ABANDA	SA_REP	2008-04-21 00:00:00	Sales
12	173	Sundita	Kumar	SKUMAR	SA_REP	2008-04-21 00:00:00	Sales
13	128	Steven	Markle	SMARKLE	ST_CLERK	2008-03-08 00:00:00	Shipping

Name: _____Skylar Trendley___________________CS 5300 Chapter 6 and 7 Homework

BELOW IS SCRIPT USED TO GENERATE ANSWERS TO QUESTIONS 1-5

```
DROP TABLE ART;
DROP TABLE ORIGIN;

CREATE TABLE ORIGIN
(
        Origin_ID       CHAR(4)             NOT NULL,
        Location        VARCHAR(25)         NOT NULL
);

ALTER TABLE ORIGIN ADD PRIMARY KEY (Origin_ID);

CREATE TABLE ART
(
        Item            CHAR(4)             NOT NULL,
        Origin_ID       CHAR(4)     NOT NULL,
        Name            VARCHAR(35),
        Artist          VARCHAR(35),
        Dating          VARCHAR(25),
        Media           VARCHAR(15)
);

ALTER TABLE ART ADD PRIMARY KEY (Item);
ALTER TABLE ART
        ADD CONSTRAINT fk_origin
                FOREIGN KEY(Origin_ID)
                REFERENCES ORIGIN(Origin_ID)
                ON DELETE CASCADE;

GRANT SELECT ON ART TO KOOBP;
GRANT SELECT ON ORIGIN TO KOOBP;


INSERT ALL
        INTO ORIGIN (Origin_ID,Location) VALUES('1111','China')
        INTO ORIGIN (Origin_ID,Location) VALUES('3543','France')
        INTO ORIGIN (Origin_ID,Location) VALUES('6943','Japan')
        INTO ORIGIN (Origin_ID,Location) VALUES('8415','Sri Lanka')
        INTO ORIGIN (Origin_ID,Location) VALUES('1598','Tajikistan')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('9182','One
Hundred Horses','Lang Shining','1111','960 to 1127','Painting')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('6922','The Great
Wave off Kanagawa','Katsushika Hokusai','6943','1829 to 1833','Painting')
```

```
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('2049','Toluvila
Statue','NULL','8415','300 to 400','Statuary')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('2038','Sasanian
Silver Vessel','NULL','1598','700 to 722','Silver')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('3964','Nymph of
the Luo River','Gu Kaizhi','1111','317 to 420','Painting')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('3097','The Hunt
of the Unicorn','NULL','3543','1680','Tapestries')
        SELECT * FROM DUAL;

SELECT * FROM ORIGIN;
SELECT * FROM ART;

/*
DELETE
FROM
        ORIGIN
WHERE
        LOCATION = 'China';
*/
SELECT COUNT(Item)
FROM ART;

DELETE
FROM ART;

INSERT ALL
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('9182','The
Thinker','Auguste Rodin','3543','1902','Statuary')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('6922','Nine
Dragons','Chen Rong','1111','1244','Painting')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('2049','Old
Plum','Kano Sansetsu','6943','1646','Painting')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('2038','Put Down
Your Whip','Xu Beihong','3543','1939','Painting')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media) VALUES('3964','The Great
Wave off Kanagawa','Hokusai','6943','1829 to 1832','Woodblock Print')
        INTO ART (Item,Name,Artist,Origin_ID,Dating,Media)
VALUES('3097','Apocalypse Tapestry','Jean Bondol, Nicolas Bataille','3543','1377 to
1382','Tapestries')
        SELECT * FROM DUAL;

SELECT * FROM ART;
```

```
ALTER TABLE ORIGIN
    ADD
    (
        COUNTRY VARCHAR(35),
        CITY VARCHAR(35)
    );


UPDATE ORIGIN
    SET COUNTRY=LOCATION;

ALTER TABLE ORIGIN
    DROP COLUMN LOCATION;

SELECT * FROM ORIGIN;
```