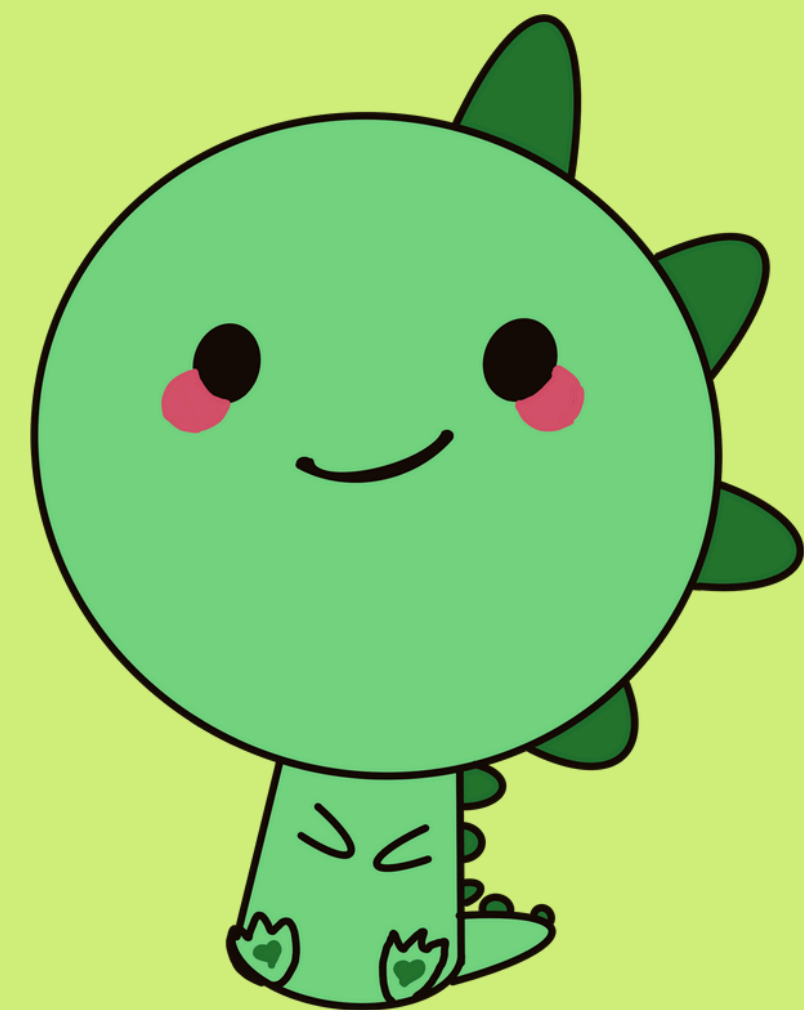
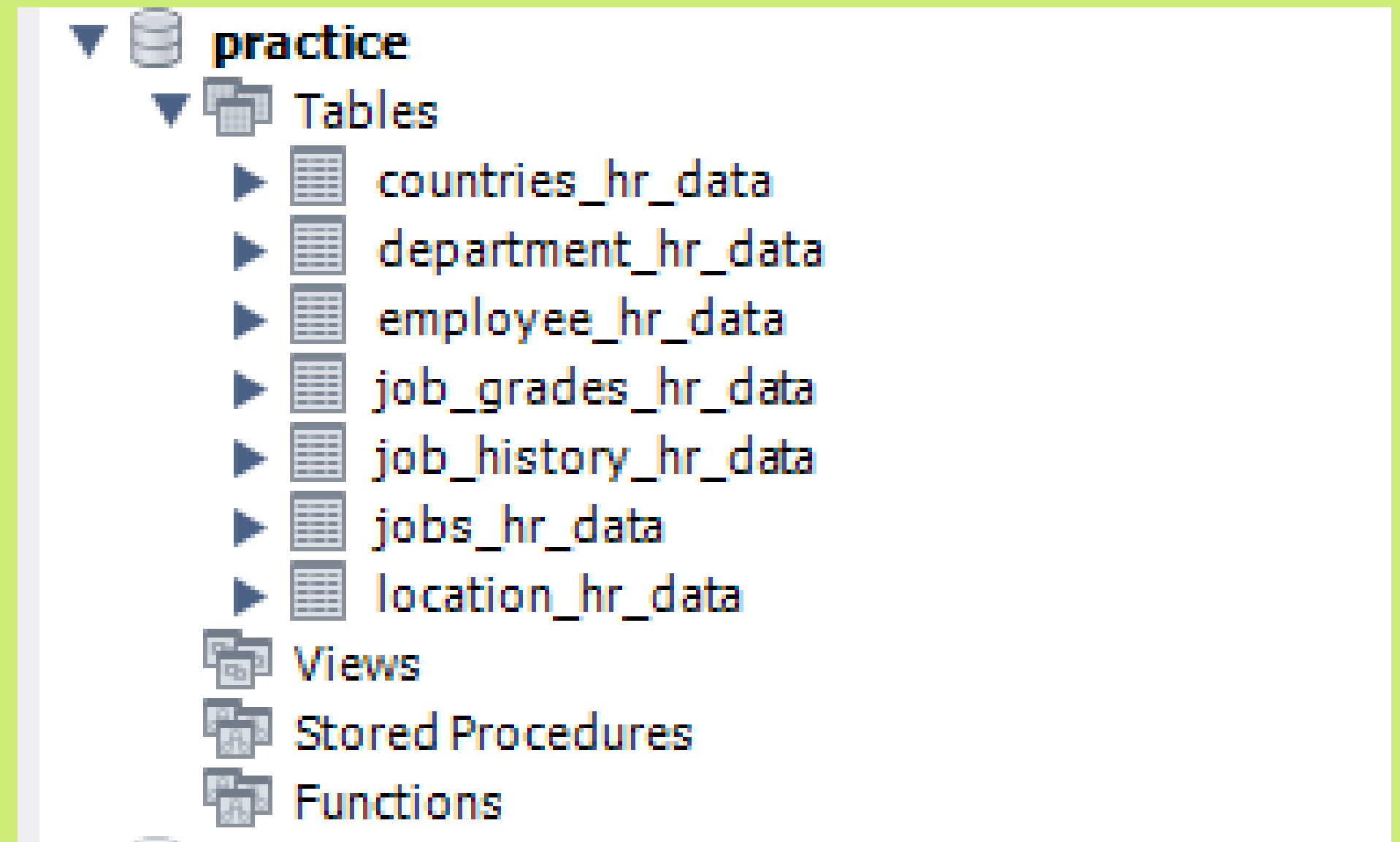


PRACTICING SQL QUERIES



CREATE DATABASE PRACTICE;

AND UPLOAD ALL THE TABLES



I. WRITE A SQL QUERY TO FIND THOSE EMPLOYEES WHOSE SALARY IS HIGHER THAN 9000. RETURN FIRST NAME, LAST NAME AND DEPARTMENT NUMBER AND SALARY

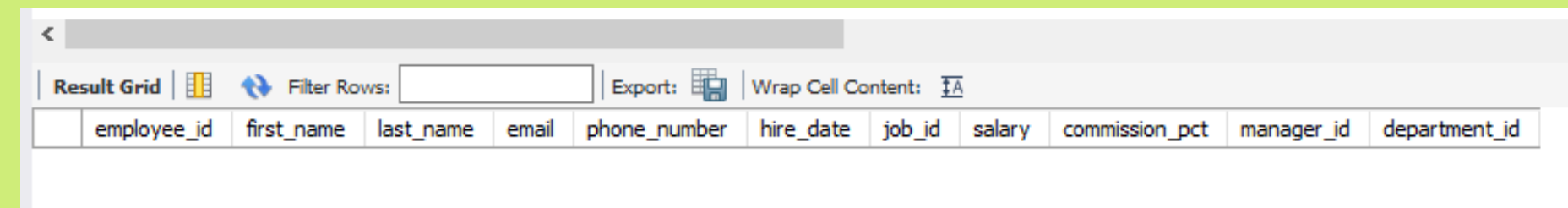
```
select * from employee_hr_data
```

```
select  
first_name,last_name,department_id  
from employee_hr_data where salary  
>9000
```

	first_name	last_name	department_id
►	Steven	King	90
	Neena	Kochhar	90
	Lex	De Haan	90
	Nancy	Greenberg	100
	Den	Raphaely	30
	John	Russell	80
	Karen	Partners	80
	Alberto	Errazuriz	80
	Gerald	Cambrault	80
	Eleni	Zlotkey	80
	Peter	Tucker	80
	David	Bernstein	80
	Janette	King	80
	Patrick	Sully	80

2 WRITE A SQL QUERY TO IDENTIFY EMPLOYEES WHO DO NOT HAVE A DEPARTMENT NUMBER. RETURN EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION_PCT, MANAGER_ID AND DEPARTMENT_ID

```
SELECT  
EMPLOYEE_ID, FIRST_NAME, LAST  
_NAME, EMAIL, PHONE_NUMBER,  
HIRE_DATE,  
JOB_ID, SALARY, COMMISSION_  
PCT, MANAGER_ID ,  
DEPARTMENT_ID  
FROM EMPLOYEE_HR_DATA  
WHERE DEPARTMENT_ID = ' ';
```

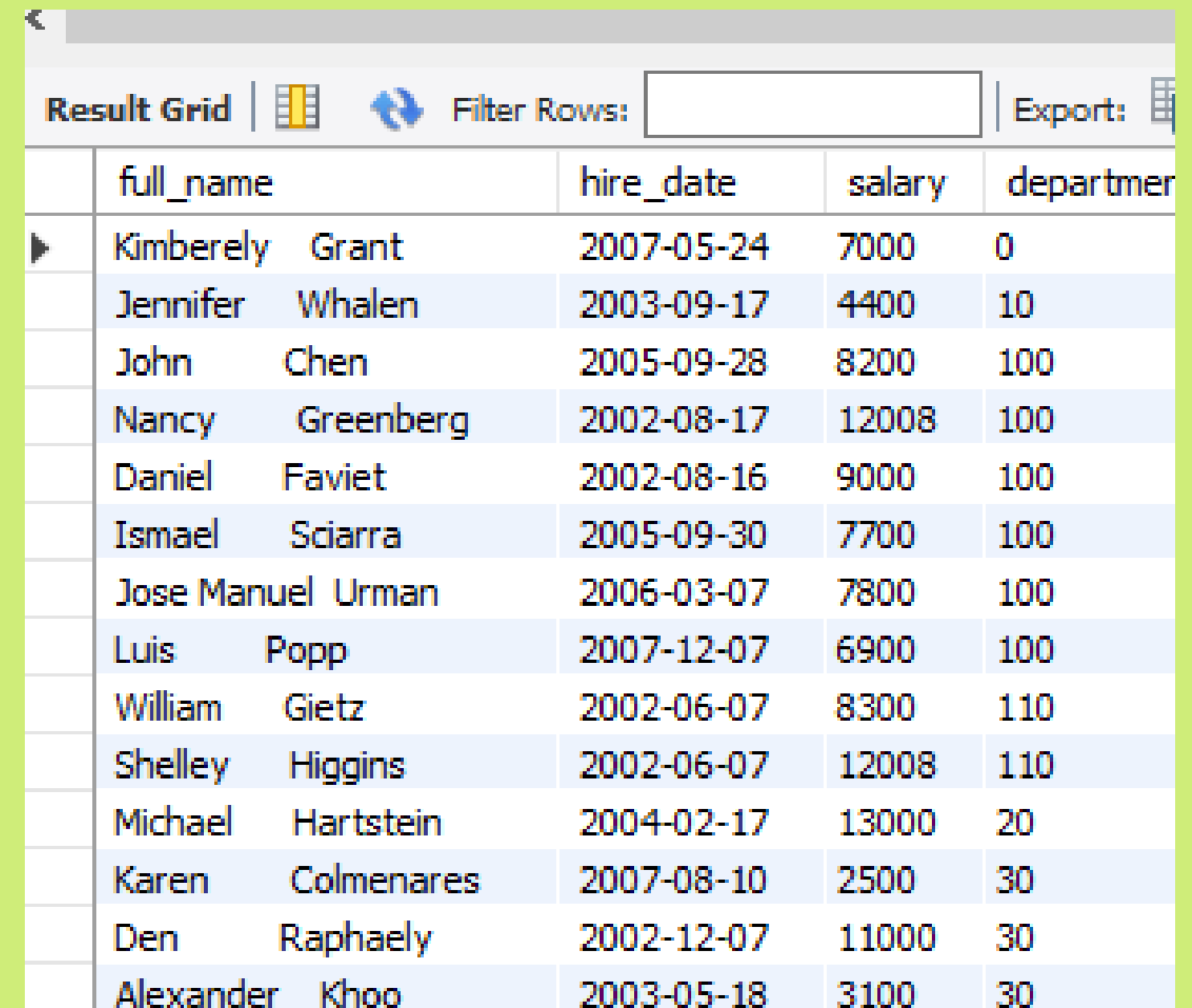


The screenshot shows a database query result grid interface. At the top, there is a toolbar with a back arrow, a 'Result Grid' button, a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' button. Below the toolbar is a table with 11 columns: employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, and department_id. The table is currently empty.

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
-------------	------------	-----------	-------	--------------	-----------	--------	--------	----------------	------------	---------------

3. WRITE A SQL QUERY TO FIND THOSE EMPLOYEES WHOSE FIRST NAME DOES NOT CONTAIN THE LETTER 'T'. SORT THE RESULT-SET IN ASCENDING ORDER BY DEPARTMENT ID. RETURN FULL NAME (FIRST AND LAST NAME TOGETHER), HIRE_DATE, SALARY AND DEPARTMENT_ID.

```
SELECT
CONCAT(FIRST_NAME, LAST_N
AME) AS
FULL_NAME, HIRE_DATE, SALAR
Y, DEPARTMENT_ID FROM
EMPLOYEE_HR_DATA WHERE
FIRST_NAME NOT LIKE '%T%'
ORDER BY DEPARTMENT_ID;
```







The screenshot shows a database query result grid with the following columns: full_name, hire_date, salary, and department_id. The data is sorted by department_id in ascending order. The first row is highlighted with a mouse cursor.

	full_name	hire_date	salary	department_id
▶	Kimberely Grant	2007-05-24	7000	0
	Jennifer Whalen	2003-09-17	4400	10
	John Chen	2005-09-28	8200	100
	Nancy Greenberg	2002-08-17	12008	100
	Daniel Faviot	2002-08-16	9000	100
	Ismael Sciarra	2005-09-30	7700	100
	Jose Manuel Urman	2006-03-07	7800	100
	Luis Popp	2007-12-07	6900	100
	William Gietz	2002-06-07	8300	110
	Shelley Higgins	2002-06-07	12008	110
	Michael Hartstein	2004-02-17	13000	20
	Karen Colmenares	2007-08-10	2500	30
	Den Raphaely	2002-12-07	11000	30
	Alexander Khoo	2003-05-18	3100	30

4. WRITE A SQL QUERY TO FIND THOSE EMPLOYEES WHO EARN BETWEEN 9000 AND 12000 (BEGIN AND END VALUES ARE INCLUDED.) AND GET SOME COMMISSION.
RETURN ALL FIELDS

```
SELECT * FROM  
EMPLOYEE_HR_DATA  
WHERE SALARY >=9000  
AND SALARY <=12000 AND  
COMMISSION_PCT>0 ;
```

result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 									
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID
147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	2005-03-10	SA_MAN	12000	0.3	100
148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	2007-10-15	SA_MAN	11000	0.3	100
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	2008-01-29	SA_MAN	10500	0.2	100
150	Peter	Tucker	PTUCKER	011.44.1344.129268	2005-01-30	SA_REP	10000	0.3	145
151	David	Bernstein	DBERNSTE	011.44.1344.345268	2005-03-24	SA_REP	9500	0.25	145
152	Peter	Hall	PHALL	011.44.1344.478968	2005-08-20	SA_REP	9000	0.25	145
156	Janette	King	JKING	011.44.1345.429268	2004-01-30	SA_REP	10000	0.35	146
157	Patrick	Sully	PSULLY	011.44.1345.929268	2004-03-04	SA_REP	9500	0.35	146
158	Allan	McEwen	AMCEWEN	011.44.1345.829268	2004-08-01	SA_REP	9000	0.35	146

5. WRITE A SQL QUERY TO FIND THOSE EMPLOYEES WHO DO NOT EARN ANY COMMISSION. RETURN FULL NAME (FIRST AND LAST NAME), AND SALARY

```
SELECT
CONCAT(TRIM(FIRST_NAME)
,TRIM(LAST_NAME)) AS
FULL_NAME ,SALARY FROM
EMPLOYEE_HR_DATA WHERE
COMMISSION_PCT=0;
```

Result Grid			Filter Rows:
	full_name	salary	
▶	FIRST_NAME LAST_NAME	SALARY	
	StevenKing	24000	
	NeenaKochhar	17000	
	LexDe Haan	17000	
	AlexanderHunold	9000	
	BruceErnst	6000	
	DavidAustin	4800	
	ValliPataballa	4800	
	DianaLorentz	4200	
	NancyGreenberg	12008	
	DanielFaviet	9000	
	JohnChen	8200	
	IsmaelSciarra	7700	

6. WRITE A SQL QUERY TO FIND THOSE EMPLOYEES WHO WORK UNDER A MANAGER.
RETURN FULL NAME (FIRST AND LAST NAME), SALARY, AND MANAGER ID

```
SELECT  
CONCAT(FIRST_NAME, LAST  
_NAME) AS  
FULL_NAME, SALARY ,  
MANAGER_ID FROM  
EMPLOYEE_HR_DATA WHERE  
MANAGER_ID != 0;
```

	full_name	salary	manager_id
►	Neena Kochhar	17000	100
	Lex De Haan	17000	100
	Alexander Hunold	9000	102
	Bruce Ernst	6000	103
	David Austin	4800	103
	Valli Pataballa	4800	103
	Diana Lorentz	4200	103
	Nancy Greenberg	12008	101
	Daniel Faviet	9000	108
	John Chen	8200	108
	Ismael Sciarra	7700	108
	Jose Manuel Urman	7800	108
	Luis Popp	6900	108
	Den Raphaely	11000	100

7. WRITE A SQL QUERY TO FIND EMPLOYEES WHOSE FIRST NAMES CONTAIN THE LETTERS F, T, OR M. SORT THE RESULT-SET IN DESCENDING ORDER BY SALARY. RETURN ALL FIELDS


```
SELECT * FROM
EMPLOYEE_HR_DATA WHERE
FIRST_NAME LIKE '%F%' OR
FIRST_NAME LIKE '%T%' OR
FIRST_NAME LIKE '%M%' ORDER
BY SALARY DESC;
```


	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
▶	170	Tayler	Fox	TFOX	011.44.1343.729268	2006-01-24	SA_REP
	157	Patrick	Sully	PSULLY	011.44.1345.929268	2004-03-04	SA_REP
	152	Peter	Hall	PHALL	011.44.1344.478968	2005-08-20	SA_REP
	176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	2006-03-24	SA_REP
	206	William	Gietz	WGIETZ	515.123.8181	2002-06-07	AC_ACCOUNT
	121	Adam	Fripp	AFRIPP	650.123.2234	2005-04-10	ST_MAN
	120	Matthew	Weiss	MWEISS	650.123.1234	2004-07-18	ST_MAN
	153	Christopher	Olsen	COLSEN	011.44.1344.498718	2006-03-30	SA_REP
	122	Payam	Kaufling	PKAUFLIN	650.123.3234	2003-05-01	ST_MAN
	112	Jose Manuel	Urman	JMURMAN	515.124.4469	2006-03-07	FI_ACCOUNT
	111	Ismael	Sciarra	ISCIARRA	515.124.4369	2005-09-30	FI_ACCOUNT
	154	Nanette	Cambrault	NCAMBRAU	011.44.1344.987668	2006-12-09	SA_REP
	171	William	Smith	WSMITH	011.44.1343.629268	2007-02-23	SA_REP

8. WRITE A SQL QUERY TO FIND THOSE EMPLOYEES WHO EARN ABOVE 12000 OR THE SEVENTH CHARACTER IN THEIR PHONE NUMBER IS 3. SORT THE RESULT-SET IN DESCENDING ORDER BY FIRST NAME. RETURN FULL NAME (FIRST NAME AND LAST NAME), HIRE DATE, COMMISSION PERCENTAGE, EMAIL, AND TELEPHONE SEPARATED BY '-', AND SALARY

```
SELECT
CONCAT(FIRST_NAME, LAST_NAME) AS
FULL_NAME, HIRE_DATE, COMMISSION_P
CT, EMAIL, SALARY, PHONE_NUMBER FROM
EMPLOYEE_HR_DATA WHERE
SALARY > 12000 OR PHONE_NUMBER LIKE
'%_____3%' ORDER BY FIRST_NAME
DESC;
```


result Grid






Filter Rows:

Export:



Wrap Cell Content:



full_name	hire_date	commission_pct	email	salary	PHONE_NUMBER
William Smith	2007-02-23	0.15	WSMITH	7400	011.44.1343.629268
William Gietz	2002-06-07	0	WGIETZ	8300	515.123.8181
Valli Pataballa	2006-02-05	0	VPATABAL	4800	590.423.4560
TJ Olson	2007-04-10	0	TJOLSON	2100	650.124.8234
Timothy Gates	2006-07-11	0	TGATES	2900	650.505.3876
Tayler Fox	2006-01-24	0.2	TFOX	9600	011.44.1343.729268
Susan Mavris	2002-06-07	0	SMAVRIS	6500	515.123.7777
Sundita Kumar	2008-04-21	0.1	SKUMAR	6100	011.44.1343.329268
Sundar Aude	2008-03-24	0.1	SANDE	6400	011.44.1346.629268
Steven King	2003-06-17	0	SKING	24000	515.123.4567
Steven Markle	2008-03-08	0	SMARKLE	2200	650.124.1434
Stephen Stiles	2005-10-26	0	SSTILES	3200	650.121.2034
Shelli Baida	2005-12-24	0	SBAIDA	2900	515.127.4563
Shelley Higgins	2002-06-07	0	SHIGGINS	12008	515.123.8080

9. WRITE A SQL QUERY TO FIND THOSE EMPLOYEES WHOSE FIRST NAME CONTAINS A CHARACTER 'S' IN THE THIRD POSITION. RETURN FIRST NAME, LAST NAME AND DEPARTMENT ID.

```
SELECT  
FIRST_NAME, LAST_NAME,  
DEPARTMENT_ID FROM  
EMPLOYEE_HR_DATA  
WHERE FIRST_NAME LIKE  
'%S%';
```

Result Grid			
Filter Rows: <input type="text"/>			
Export: <input type="text"/>			
	first_name	last_name	department_id
▶	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
	Steven	King	90
	Ismael	Sciarra	100
	Jose Manuel	Urman	100
	Luis	Popp	100
	Shelli	Baida	30
	Sigal	Tobias	30
	Shanta	Vollman	50
	James	Landry	50
	Steven	Markle	50
	James	Marlow	50
	Jason	Mallin	50
	Renske	Ladwig	50

10. WRITE A SQL QUERY TO COUNT THE NUMBER OF EMPLOYEES, THE SUM OF ALL SALARY, AND DIFFERENCE BETWEEN THE HIGHEST SALARY AND LOWEST SALARIES BY EACH JOB ID. RETURN JOB_ID,COUNT, SUM, SALARY_DIFFERENCE

```
SELECT  
JOB_ID,COUNT(EMPLOYEE_ID) AS EMP , SUM(SALARY) AS  
SAL, MAX(SALARY) -  
MIN(SALARY) AS DIFF FROM  
EMPLOYEE_HR_DATA GROUP  
BY JOB_ID ;
```

job_id	emp	sal	diff
JOB_ID	1	0	0
AD_PRES	1	24000	0
AD_VP	2	34000	0
IT_PROG	5	28800	4800
FI_MGR	1	12008	0
FI_ACCOUNT	5	39600	2100
PU_MAN	1	11000	0
PU_CLERK	5	13900	600
ST_MAN	5	36400	2400
ST_CLERK	20	55700	1500
SA_MAN	5	61000	3500
SA_REP	30	250500	-400
SH_CLERK	20	64300	1700

II. WRITE A SQL QUERY TO FIND EACH JOB IDS WHERE TWO OR MORE EMPLOYEES WORKED FOR MORE THAN 300 DAYS. RETURN JOB ID

```
SELECT
  JOB_ID
FROM
  EMPLOYEE_HR_DATA
GROUP BY
  JOB_ID
HAVING
  COUNT(CASE WHEN
    DATEDIFF(CURDATE(), HIRE_DATE)
    > 300 THEN 1 END) >= 2;
```

	job_id
▶	AD_VP
	IT_PROG
	FI_ACCOUNT
	PU_CLERK
	ST_MAN
	ST_CLERK
	SA_MAN
	SA_REP
	SH_CLERK

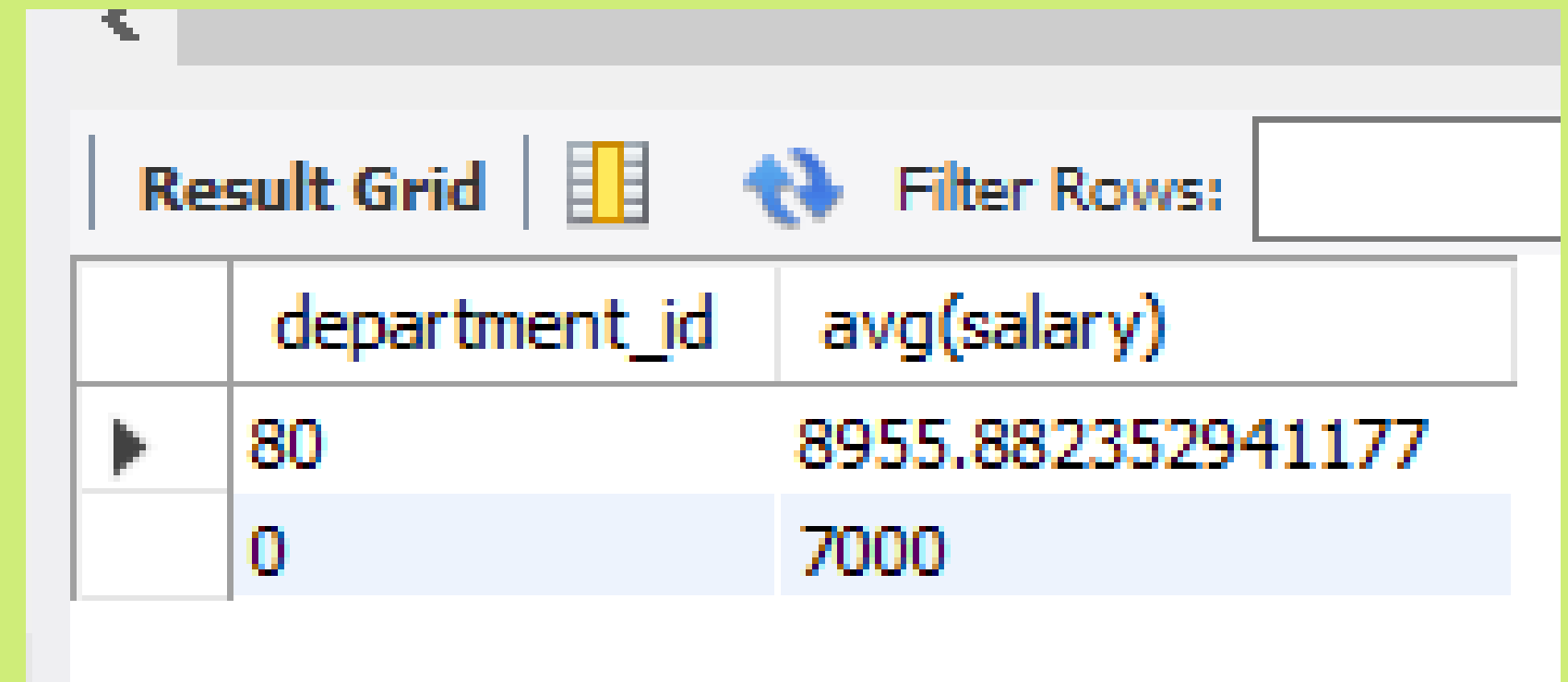
12. WRITE A SQL QUERY TO COUNT THE NUMBER OF EMPLOYEES WORKED UNDER EACH MANAGER. RETURN MANAGER ID AND NUMBER OF EMPLOYEES

```
SELECT  
MANAGER_ID,COUNT(EMPLOYEE_ID) AS COUNTEMP  
FROM  
EMPLOYEE_HR_DATA  
GROUP BY MANAGER_ID;
```

	manager_id	countemp
▶	MANAGER_ID	1
	0	1
	100	14
	102	1
	103	4
	101	5
	108	5
	114	5
	120	8
	121	8
	122	8
	123	8
	124	8
	145	6

13. WRITE A SQL QUERY TO CALCULATE THE AVERAGE SALARY OF EMPLOYEES WHO RECEIVE A COMMISSION PERCENTAGE FOR EACH DEPARTMENT. RETURN DEPARTMENT ID, AVERAGE SALARY.

```
SELECT  
DEPARTMENT_ID,AVG(SALAR  
Y) FROM  
EMPLOYEE_HR_DATA WHERE  
COMMISSION_PCT !=0  
GROUP BY DEPARTMENT_ID ;
```

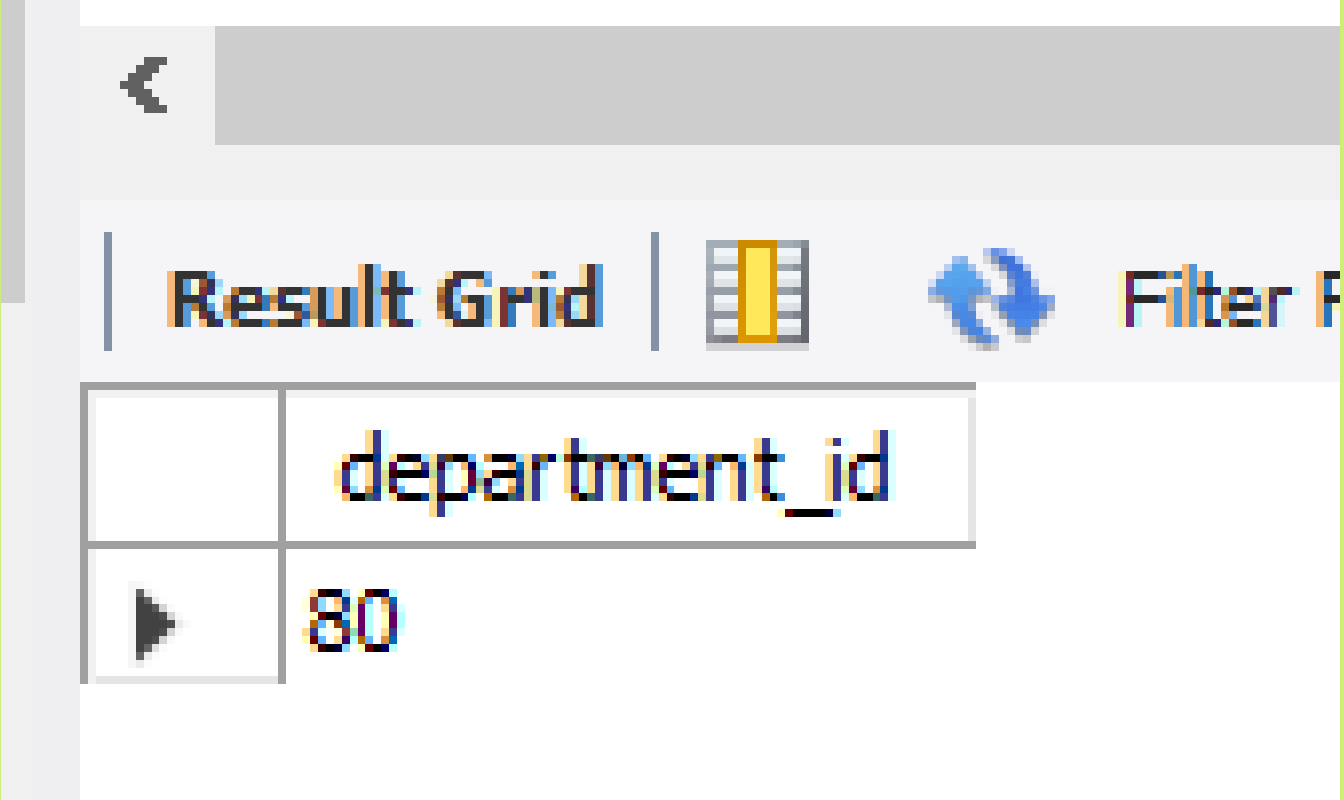


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains two columns: 'department_id' and 'avg(salary)'. There are two rows of data. The first row has a department_id of 80 and an average salary of 8955.882352941177. The second row has a department_id of 0 and an average salary of 7000. The second row is highlighted in light blue.

	department_id	avg(salary)
▶	80	8955.882352941177
	0	7000

14. WRITE A SQL QUERY TO FIND THE DEPARTMENTS WHERE MORE THAN TEN EMPLOYEES RECEIVE COMMISSIONS. RETURN DEPARTMENT ID

```
SELECT DEPARTMENT_ID FROM  
EMPLOYEE_HR_DATA WHERE  
COMMISSION_PCT !=0 GROUP BY  
DEPARTMENT_ID HAVING  
COUNT(*)>10;
```

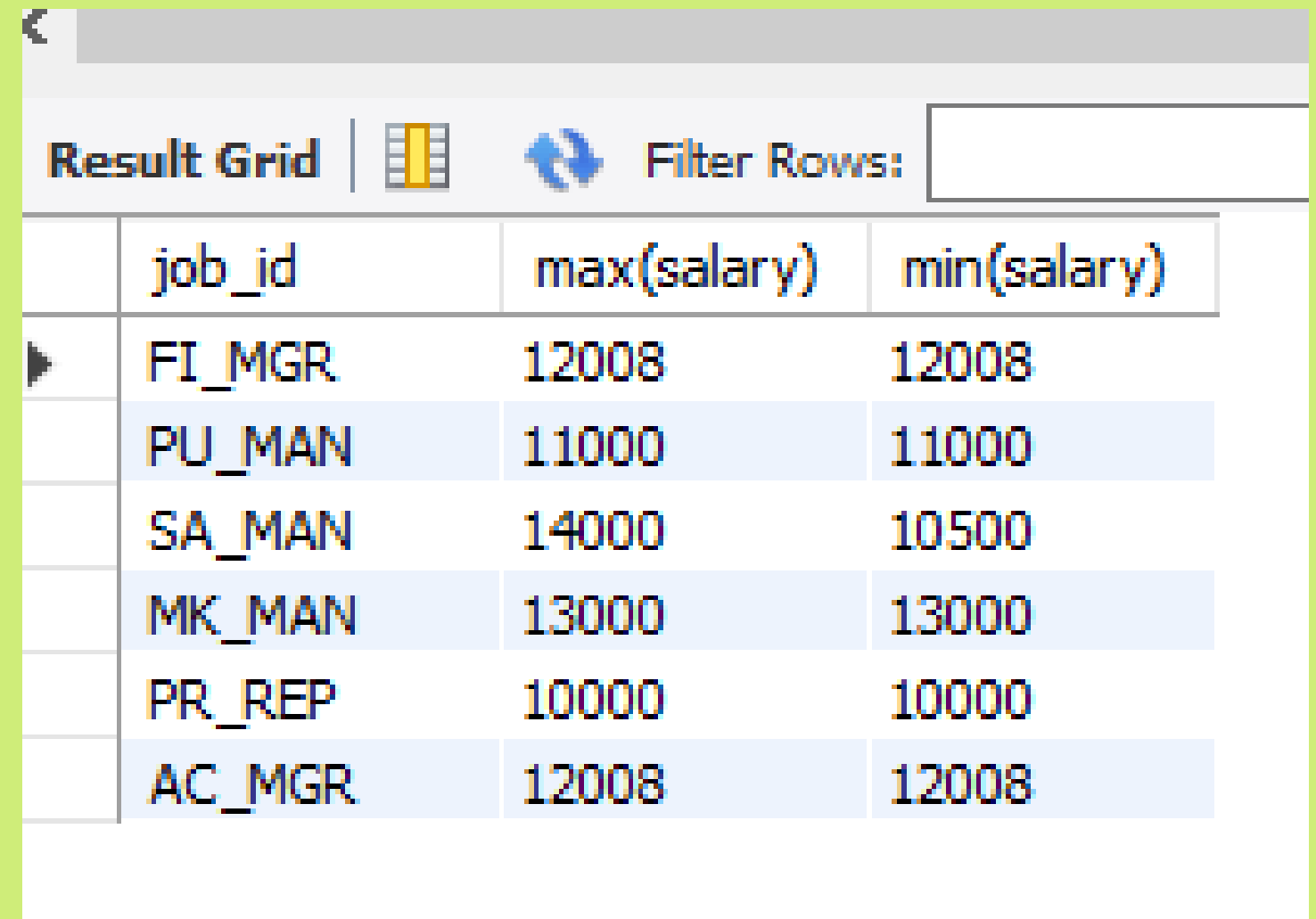


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one row with the column header 'department_id' and the value '80'. There are navigation icons and a 'Filter' button visible above the grid.

	department_id
▶	80

15. WRITE A SQL QUERY TO FIND THOSE JOB TITLES WHERE MAXIMUM SALARY FALLS BETWEEN 10000 AND 15000 (BEGIN AND END VALUES ARE INCLUDED).
RETURN JOB_TITLE, MAX_SALARY✕MIN_SALARY

```
SELECT JOB_ID,  
MAX(SALARY),MIN(SALARY)  
FROM EMPLOYEE_HR_DATA  
GROUP BY JOB_ID HAVING  
MAX(SALARY) BETWEEN 10000  
AND 15000;
```




The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a SQL query. The grid has four columns: 'job_id', 'max(salary)', and 'min(salary)'. There are six rows of data. The first row is 'FI_MGR' with a max salary of 12008 and a min salary of 12008. The second row is 'PU_MAN' with a max salary of 11000 and a min salary of 11000. The third row is 'SA_MAN' with a max salary of 14000 and a min salary of 10500. The fourth row is 'MK_MAN' with a max salary of 13000 and a min salary of 13000. The fifth row is 'PR_REP' with a max salary of 10000 and a min salary of 10000. The sixth row is 'AC_MGR' with a max salary of 12008 and a min salary of 12008. The interface also includes a 'Filter Rows' button and a search bar.

	job_id	max(salary)	min(salary)
▶	FI_MGR	12008	12008
	PU_MAN	11000	11000
	SA_MAN	14000	10500
	MK_MAN	13000	13000
	PR_REP	10000	10000
	AC_MGR	12008	12008

16. WRITE A SQL QUERY TO FIND DETAILS OF THOSE JOBS WHERE THE MINIMUM SALARY EXCEEDS 9000. RETURN ALL THE FIELDS OF JOBS

```
SELECT JOB_ID , MIN(SALARY)
FROM EMPLOYEE_HR_DATA
GROUP BY JOB_ID HAVING
MIN(SALARY)>9000;
```

Result Grid			 Filter Rows: <input type="text"/>	
	job_id	min(salary)		
	AD_PRES	24000		
	AD_VP	17000		
	FI_MGR	12008		
	PU_MAN	11000		
	SA_MAN	10500		
	SA_REP	10000		
	MK_MAN	13000		
	PR_REP	10000		
	AC_MGR	12008		

17. WRITE A SQL QUERY TO FIND THOSE EMPLOYEES WHO WORK IN THE SAME DEPARTMENT AS 'CLARA'. EXCLUDE ALL THOSE RECORDS WHERE FIRST NAME IS 'CLARA'. RETURN FIRST NAME, LAST NAME AND HIRE DATE.

```
set @a = (select department_id from  
employee_hr_data where first_name like  
'%Clara%')
```


```
select @a
```

```
select first_name , last_name ,hire_date  
from employee_hr_data where  
department_id = @a
```

	first_name	last_name	hire_date
▶	John	Russell	2004-10-01
	Karen	Partners	2005-01-05
	Alberto	Errazuriz	2005-03-10
	Gerald	Cambrault	2007-10-15
	Eleni	Zlotkey	2008-01-29
	Peter	Tucker	2005-01-30
	David	Bernstein	2005-03-24
	Peter	Hall	2005-08-20
	Christopher	Olsen	2006-03-30
	Nanette	Cambrault	2006-12-09
	Oliver	Tuvault	2007-11-23
	Janette	King	2004-01-30
	Patrick	Sully	2004-03-04
	Allan	McEwen	2004-08-01

18. WRITE A SQL QUERY TO FIND THOSE EMPLOYEES WHO EARN MORE THAN THE AVERAGE SALARY AND WORK IN THE SAME DEPARTMENT AS AN EMPLOYEE WHOSE FIRST NAME CONTAINS THE LETTER 'J'. RETURN EMPLOYEE ID, FIRST NAME AND SALARY


```
WITH CTE AS (  
    SELECT DEPARTMENT_ID  
    FROM EMPLOYEE_HR_DATA  
    WHERE FIRST_NAME LIKE '%J%'  
    ),  
    CTEI AS (  
    SELECT DEPARTMENT_ID, EMPLOYEE_ID, FIRST_NAME,  
           SALARY  
    FROM EMPLOYEE_HR_DATA  
    WHERE SALARY > (SELECT AVG(SALARY) FROM  
                     EMPLOYEE_HR_DATA)  
    )  
SELECT EMPLOYEE_ID, FIRST_NAME, SALARY FROM CTEI  
JOIN CTE ON CTEI.DEPARTMENT_ID =  
           CTE.DEPARTMENT_ID;
```


Result Grid  Filter Rows: <input type="text"/>			
	employee_id	first_name	salary
▶	113	Luis	6900
	112	Jose Manuel	7800
	111	Ismael	7700
	110	John	8200
	109	Daniel	9000
	108	Nancy	12008
	113	Luis	6900
	112	Jose Manuel	7800
	111	Ismael	7700
	110	John	8200
	109	Daniel	9000
	108	Nancy	12008
	123	Shanta	6500
	122	Payam	7900

19. WRITE A QUERY TO DISPLAY THE EMPLOYEE ID, NAME (FIRST NAME AND LAST NAME) AND THE JOB ID COLUMN WITH A MODIFIED TITLE SALESMAN FOR THOSE EMPLOYEES WHOSE JOB TITLE IS ST_MAN AND DEVELOPER FOR WHOSE JOB TITLE IS IT_PROG


```
SELECT
    EMPLOYEE_ID,
    CONCAT(FIRST_NAME, ' ', LAST_NAME) AS
        NAME,
    CASE
        WHEN TRIM(JOB_ID) = 'ST_MAN' THEN
            'SALESMAN'
        WHEN TRIM(JOB_ID) = 'IT_PROG' THEN
            'DEVELOPER'
        ELSE JOB_ID -- KEEPS THE ORIGINAL
        JOB_ID IF IT DOESN'T MATCH THE SPECIFIED
        VALUES
    END AS JOB_TITLE
FROM
    EMPLOYEE_HR_DATA;
```

Result Grid



 Filter Rows:

Export:






W

	employee_id	name		job_title
▶	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID
	100	Steven	King	AD_PRES
	101	Neena	Kochhar	AD_VP
	102	Lex	De Haan	AD_VP
	103	Alexander	Hunold	DEVELOPER
	104	Bruce	Ernst	DEVELOPER
	105	David	Austin	DEVELOPER
	106	Valli	Pataballa	DEVELOPER
	107	Diana	Lorentz	DEVELOPER
	108	Nancy	Greenberg	FI_MGR
	109	Daniel	Faviet	FI_ACCOU...
	110	John	Chen	FI_ACCOU...
	111	Ismael	Sciarra	FI_ACCOU...
	112	Jose Manuel	Urman	FI ACCOU...

-- JOINS



I. WRITE A SQL QUERY TO FIND THE FIRST NAME, LAST NAME, DEPARTMENT, CITY, AND STATE PROVINCE FOR EACH EMPLOYEE

```
SELECT
FIRST_NAME, LAST_NAME, D.DEPARTMENT_NAME, CITY, STATE_PROVINCE
FROM
EMPLOYEE_HR_DATA E JOIN
DEPARTMENT_HR_DATA D ON
E.DEPARTMENT_ID =
D.DEPARTMENT_ID
JOIN LOCATION_HR_DATA L ON
L.LOCATION_ID =
D.LOCATION_ID ;
```

result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 					
	first_name	last_name	department_name	city	STATE_PROVINCE
	Steven	King	Executive	Seattle	Washington
	Neena	Kochhar	Executive	Seattle	Washington
	Lex	De Haan	Executive	Seattle	Washington
	Alexander	Hunold	IT	Southlake	Texas
	Bruce	Ernst	IT	Southlake	Texas
	David	Austin	IT	Southlake	Texas
	Valli	Pataballa	IT	Southlake	Texas
	Diana	Lorentz	IT	Southlake	Texas
	Nancy	Greenberg	Finance	Seattle	Washington
	Daniel	Faviet	Finance	Seattle	Washington
	John	Chen	Finance	Seattle	Washington
	Ismael	Sciarra	Finance	Seattle	Washington
	Jose Manuel	Urman	Finance	Seattle	Washington
	Luis	Popp	Finance	Seattle	Washington




2. WRITE A SQL QUERY TO FIND THOSE EMPLOYEES WHOSE FIRST NAME CONTAINS THE LETTER 'Z'. RETURN FIRST NAME, LAST NAME, DEPARTMENT, CITY, AND STATE PROVINCE

```
SELECT  
FIRST_NAME, LAST_NAME, D.DEPART  
MENT_NAME, CITY, STATE_PROVINC  
E FROM EMPLOYEE_HR_DATA E  
JOIN DEPARTMENT_HR_DATA D ON  
E.DEPARTMENT_ID =  
D.DEPARTMENT_ID  
JOIN LOCATION_HR_DATA L ON  
L.LOCATION_ID = D.LOCATION_ID  
WHERE FIRST_NAME LIKE '%Z%';
```

Result Grid					
Filter Rows: <input type="text"/>					
Export:  Wrap Cell Content: 					
	first_name	last_name	department_name	city	STATE_PROVINCE
▶	Mozhe	Atkinson	Shipping	South San Francisco	California
	Hazel	Philtanker	Shipping	South San Francisco	California
	Elizabeth	Bates	Sales	Oxford	Oxford

3. WRITE A SQL QUERY TO FIND ALL EMPLOYEES WHO JOINED ON 1ST JANUARY 1993 AND LEFT ON OR BEFORE 31 AUGUST 1997. RETURN JOB TITLE, DEPARTMENT NAME, EMPLOYEE NAME, AND JOINING DATE OF THE JOB

```
SELECT
    J.JOB_TITLE,
    D.DEPARTMENT_NAME,
    CONCAT(E.FIRST_NAME, ' ',
    E.LAST_NAME) AS FULL_NAME,
    J.START_DATE
FROM
    EMPLOYEE_HR_DATA E
JOIN
    JOB_HISTORY_HR_DATA J ON
    E.EMPLOYEE_ID = J.EMPLOYEE_ID
JOIN
    DEPARTMENT_HR_DATA D ON
    J.DEPARTMENT_ID = D.DEPARTMENT_ID;
```

Result Grid   Filter Rows: Export:  Wrap Cell Conte					
	job_id	department_name	full_name		start_date
▶	AC_ACCOUNT	Accounting	Neena	Kochhar	1997-09-21
	AC_MGR	Accounting	Neena	Kochhar	2001-10-28
	IT_PROG	IT	Lex	De Haan	2001-01-13
	ST_CLERK	Shipping	Den	Raphaely	2006-03-24
	ST_CLERK	Shipping	Payam	Kaufling	2007-01-01
	SA_REP	Sales	Jonathon	Taylor	2006-03-24
	SA_MAN	Sales	Jonathon	Taylor	2007-01-01




4. WRITE A SQL QUERY TO FIND THE DEPARTMENT NAME AND THE FULL NAME (FIRST AND LAST NAME) OF THE MANAGER DEPARTMENT_HR_DATA

```
SELECT
DISTINCT(CONCAT(E.FIRST_NAME ,
E.LAST_NAME)),DEPARTMENT_NAME
FROM EMPLOYEE_HR_DATA E JOIN
EMPLOYEE_HR_DATA EI ON
E.EMPLOYEE_ID = EI.MANAGER_ID JOIN
DEPARTMENT_HR_DATA D ON
D.DEPARTMENT_ID = E.DEPARTMENT_ID
;
```

Result Grid			Filter Rows:	Export
	(concat(e.first_name , e.last_name))	department_name		
▶	Steven King	Executive		
	Lex De Haan	Executive		
	Alexander Hunold	IT		
	Neena Kochhar	Executive		
	Nancy Greenberg	Finance		
	Den Raphaely	Purchasing		

5. WRITE A SQL QUERY TO FIND THE DEPARTMENT NAME, FULL NAME (FIRST AND LAST NAME) OF THE
-- MANAGER AND THEIR CITY

```
SELECT DISTINCT(CONCAT(E.FIRST_NAME ,  
E.LAST_NAME)),DEPARTMENT_NAME,CITY  
FROM EMPLOYEE_HR_DATA E JOIN  
EMPLOYEE_HR_DATA EI ON E.EMPLOYEE_ID =  
EI.MANAGER_ID JOIN DEPARTMENT_HR_DATA  
D ON D.DEPARTMENT_ID = E.DEPARTMENT_ID  
JOIN LOCATION_HR_DATA L ON  
L.LOCATION_ID = D.LOCATION_ID ;
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap C			
(concat(e.first_name , e.last_name))	department_name	city	
Steven King	Executive	Seattle	
Lex De Haan	Executive	Seattle	
Alexander Hunold	IT	Southlake	
Neena Kochhar	Executive	Seattle	
Nancy Greenberg	Finance	Seattle	
Den Raphaely	Purchasing	Seattle	

THANK YOU!

