MA RAQUEL SORIANO
HHA 507

SQL Database Programming: Section 9-1: Using GROUP BY and HAVING Clauses

Vocabulary

HAVING clause – Used to specify which groups are to be displayed; restricts groups that do not meet group criteria

GROUP BY clause – Divides the rows in a table into groups

Try It / Solve It

1. In the SQL query shown below, which of the following is true about this query?
   _____ a. Kimberly Grant would not appear in the results set.
   _____ b. The GROUP BY clause has an error because the manager_id is not listed in the SELECT clause.
   _____ c. Only salaries greater than 16001 will be in the result set.
   _____ d. Names beginning with Ki will appear after names beginning with Ko.
   _____ e. Last names such as King and Kochhar will be returned even if they don't have salaries > 16000.

```
SELECT last_name, MAX(salary)
FROM employees
WHERE last_name LIKE 'K%'
GROUP BY manager_id, last_name
HAVING MAX(salary) >16000
ORDER BY last_name DESC ;
```

<<< MORE ANSWERS CONTINUE ON THE NEXT PAGE >>>

2. Each of the following SQL queries has an error. Find the error and correct it. Use Oracle Application Express to verify that your corrections produce the desired results.

a. SELECT manager_id
FROM employees
WHERE AVG(salary) <16000
GROUP BY manager_id;

```
SELECT manager_id
FROM employees
GROUP BY manager_id
HAVING AVG(salary) < 16000;
```

Results | Explain | Describe | Saved SQL | History

MANAGER_ID
124
101
103
149
201
100
205
102

b. SELECT cd_number, COUNT(title)
FROM d_cds
WHERE cd_number < 93;

```
SELECT cd_number, COUNT(title)
FROM d_cds
WHERE cd_number < 93
GROUP BY cd_number;
```

Results | Explain | Describe | Saved SQL | History

CD_NUMBER	COUNT(TITLE)
90	1
91	1
92	1

<<< MORE ANSWERS CONTINUE ON THE NEXT PAGE >>>

c. SELECT ID, MAX(ID), artist AS Artist
FROM d_songs
WHERE duration IN('3 min', '6 min', '10 min')
HAVING ID < 50
GROUP by ID;

```
SELECT ID, artist AS Artist
FROM d_songs
WHERE duration IN ('3 min', '6 min', '10 min')
GROUP BY ID, artist
HAVING ID < 50;
```

Results | Explain | Describe | Saved SQL | History

ID	ARTIST
47	The Jubilant Trio
48	Bobby West

d. SELECT loc_type, rental_fee AS Fee
FROM d_venues
WHERE id <100
GROUP BY "Fee"
ORDER BY 2;

```
SELECT loc_type, rental_fee AS Fee
FROM d_venues
WHERE id < 100
GROUP BY loc_type, rental_fee
ORDER BY 2;
```

Results | Explain | Describe | Saved SQL | History

LOC_TYPE	FEE
National Park	400/flat fee
School Hall	75/hour

<<< MORE ANSWERS CONTINUE ON THE NEXT PAGE >>>

3. Rewrite the following query to accomplish the same result:

```
SELECT DISTINCT MAX(song_id)
FROM d_track_listings
WHERE track IN ( 1, 2, 3);
```

```
SELECT MAX(song_id)
FROM d_track_listings
WHERE track IN (1, 2, 3);
```

Results | Explain | Describe | Saved SQL | History

MAX(SONG_ID)
49

4. Indicate True or False

__TRUE__ a. If you include a group function and any other individual columns in a SELECT clause, then each individual column must also appear in the GROUP BY clause.

__FALSE__ b. You can use a column alias in the GROUP BY clause.

__FALSE__ c. The GROUP BY clause always includes a group function.

5. Write a query that will return both the maximum and minimum average salary grouped by department from the employees table.

APEX | App Builder | SQL Workshop | Team Development | Gallery | Search

SQL Commands | Schema US_A296_SQL_S18_ADMIN

Language SQL | Rows 10 | Clear Command | Find Tables | Save | Run

```
SELECT MAX(avg(salary)) AS "Max Avg Salary", MIN(avg(salary)) AS "Min Avg Salary"
FROM employees
GROUP by department_id;
```

Results | Explain | Describe | Saved SQL | History

Max Avg Salary	Min Avg Salary
19333.3333333333333333333333333333333333	3371.428571428571428571428571428571428571428571428571

1 rows returned in 0.01 seconds Download

SQL Database Programming: Section 9-2: Using ROLLUP and CUBE Operations and GROUPING SETS

Vocabulary

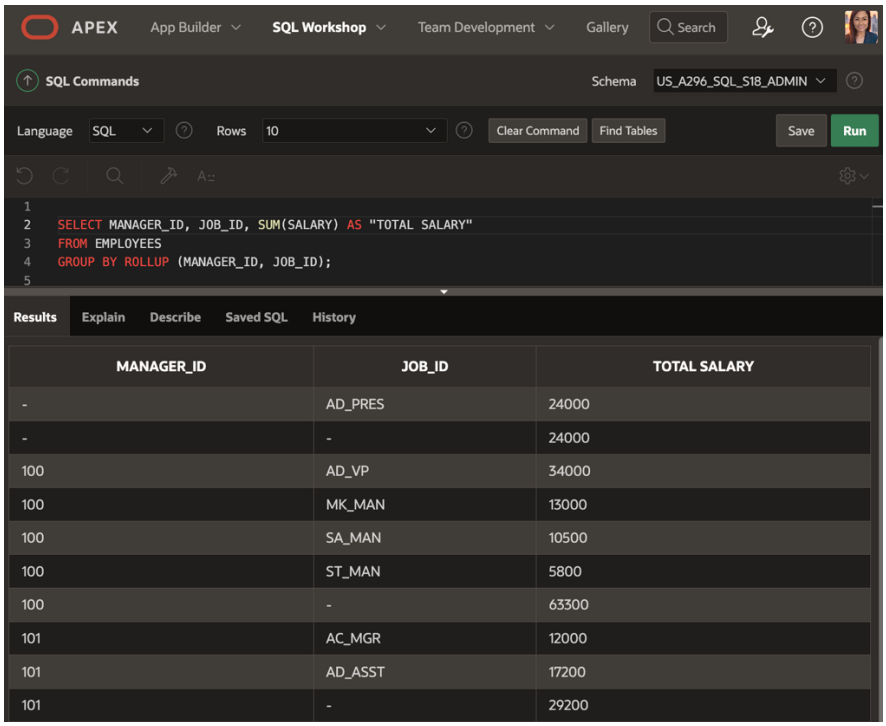
ROLLUP – Used to create subtotals that roll up from the most detailed level to a grand total, following a grouping list specified in the clause

CROSS – An extension to the GROUP BY clause like ROLLUP that produces cross-tabulation reports

GROUPING SETS – Used to specify multiple groupings of data

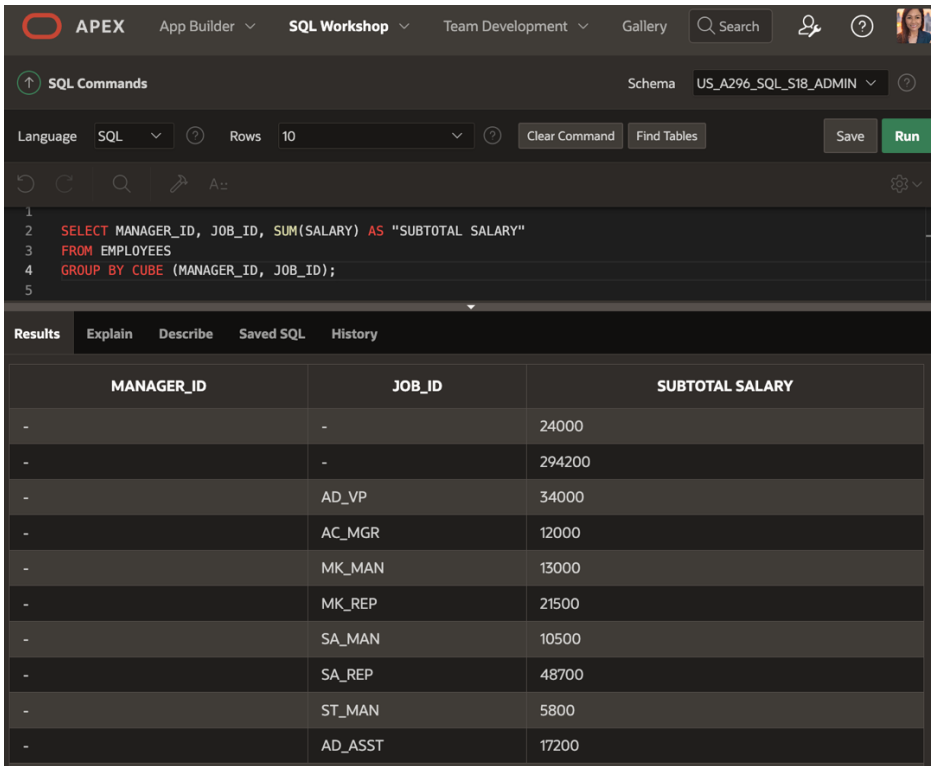
Try It / Solve It

1. Within the Employees table, each manager_id is the manager of one or more employees who each have a job_id and earn a salary. For each manager, what is the total salary earned by all of the employees within each job_id? Write a query to display the **Manager_id, job_id, and total salary**. Include in the **result the subtotal salary for each manager** and a **grand total of all salaries.**



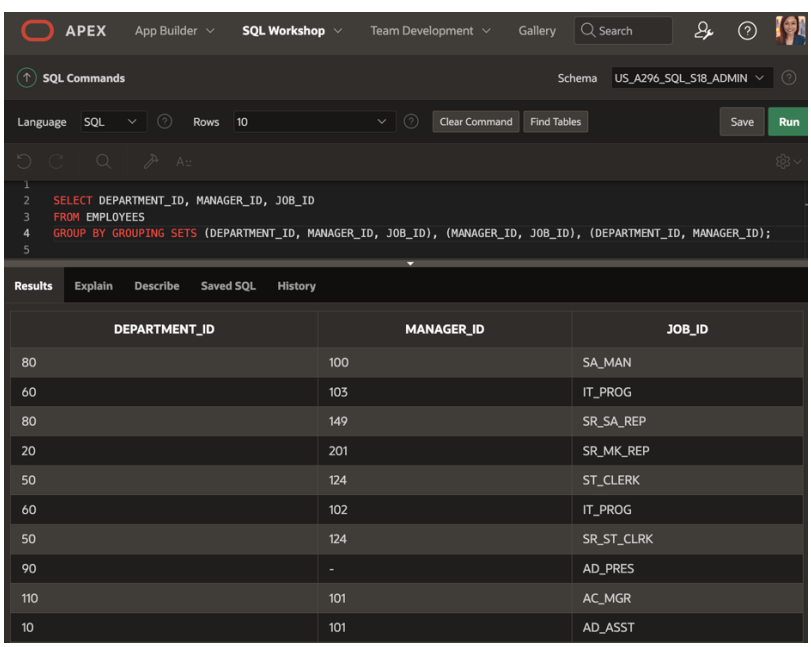
MANAGER_ID	JOB_ID	TOTAL SALARY
-	AD_PRES	24000
-	-	24000
100	AD_VP	34000
100	MK_MAN	13000
100	SA_MAN	10500
100	ST_MAN	5800
100	-	63300
101	AC_MGR	12000
101	AD_ASST	17200
101	-	29200

2. Amend the previous query to also include a **subtotal salary for each job_id** regardless of the manager_id.



MANAGER_ID	JOB_ID	SUBTOTAL SALARY
-	-	24000
-	-	294200
-	AD_VP	34000
-	AC_MGR	12000
-	MK_MAN	13000
-	MK_REP	21500
-	SA_MAN	10500
-	SA_REP	48700
-	ST_MAN	5800
-	AD_ASST	17200

3. Using GROUPING SETS, write a query to show the following groupings:
   - department_id, manager_id, job_id
   - manager_id, job_id
   - department_id, manager_id



DEPARTMENT_ID	MANAGER_ID	JOB_ID
80	100	SA_MAN
60	103	IT_PROG
80	149	SR_SA_REP
20	201	SR_MK_REP
50	124	ST_CLERK
60	102	IT_PROG
50	124	SR_ST_CLRK
90	-	AD_PRES
110	101	AC_MGR
10	101	AD_ASST

SQL Database Programming: Section 9-3: Set Operators

Vocabulary

UNION – operator that returns all rows from both tables and eliminates duplicates

SELECT LIST – columns that were made up to match queries in another table that are not in both tables

UNION ALL – operator that returns all rows from both tables, including duplicates

SET OPERATORS – used to combine results into one single result from multiple SELECT statements

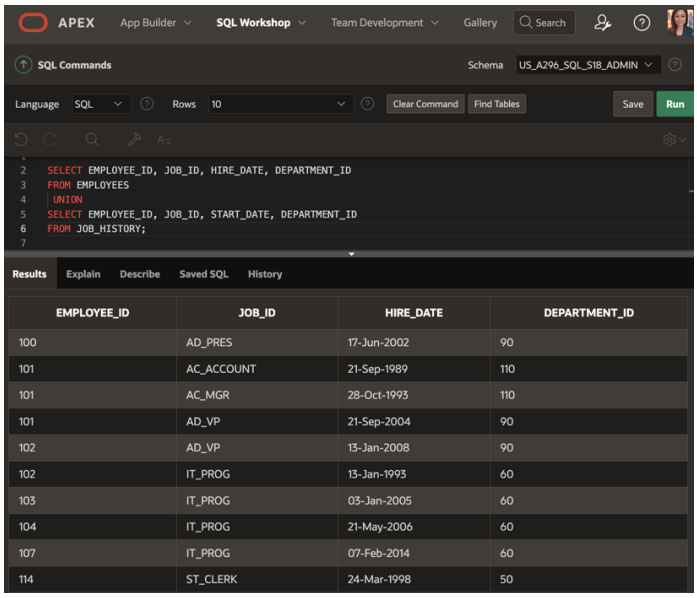
MINUS – operator that returns rows that are unique to each table

INTERSECTION – operator that returns rows common to both tables

Try It / Solve It

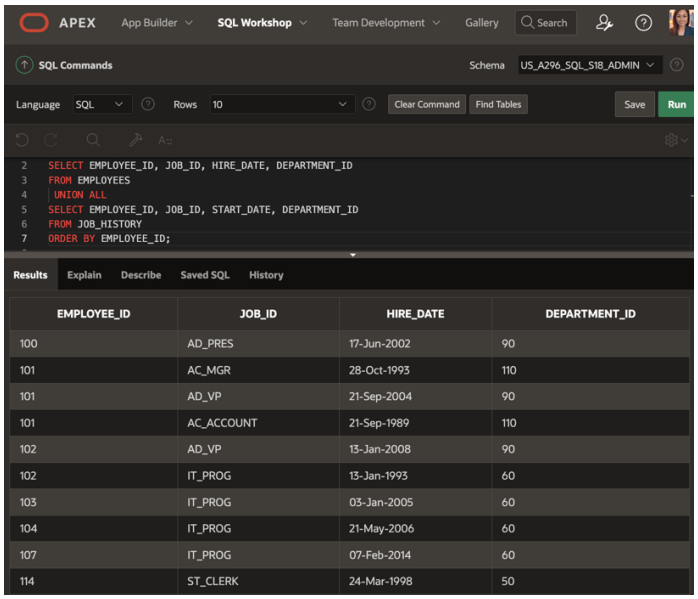
1. Name the different Set operators?
   - UNION
   - UNION ALL
   - INTERSECT
   - MINUS

2. Write <u>one query</u> to return the employee_id, job_id, hire_date, and department_id of all employees and a <u>second query</u> listing employee_id, job_id, start_date, and department_id from the job_history table and <u>combine the results as one single output.</u> Make sure you <u>suppress duplicates</u> in the output.



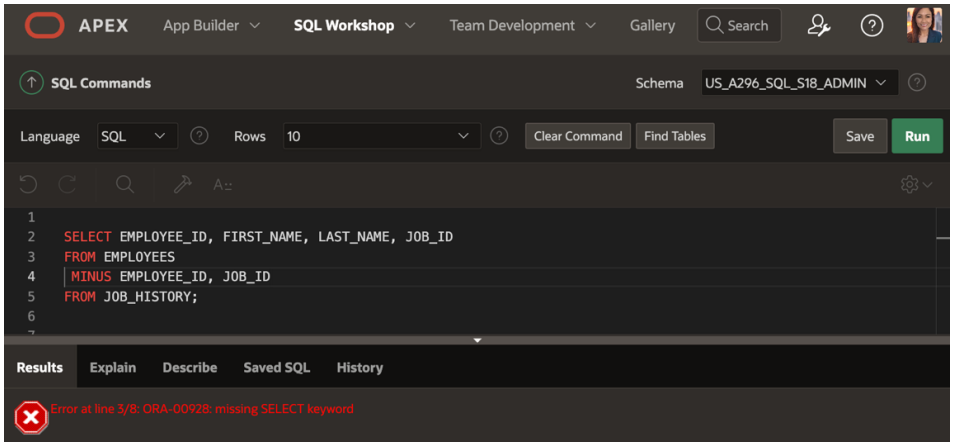
EMPLOYEE_ID	JOB_ID	HIRE_DATE	DEPARTMENT_ID
100	AD_PRES	17-Jun-2002	90
101	AC_ACCOUNT	21-Sep-1989	110
101	AC_MGR	28-Oct-1993	110
101	AD_VP	21-Sep-2004	90
102	AD_VP	13-Jan-2008	90
102	IT_PROG	13-Jan-1993	60
103	IT_PROG	03-Jan-2005	60
104	IT_PROG	21-May-2006	60
107	IT_PROG	07-Feb-2014	60
114	ST_CLERK	24-Mar-1998	50

3. Amend the previous statement to not suppress duplicates and examine the output. How many extra rows did you get returned and which were they? Sort the output by employee_id to make it easier to spot.

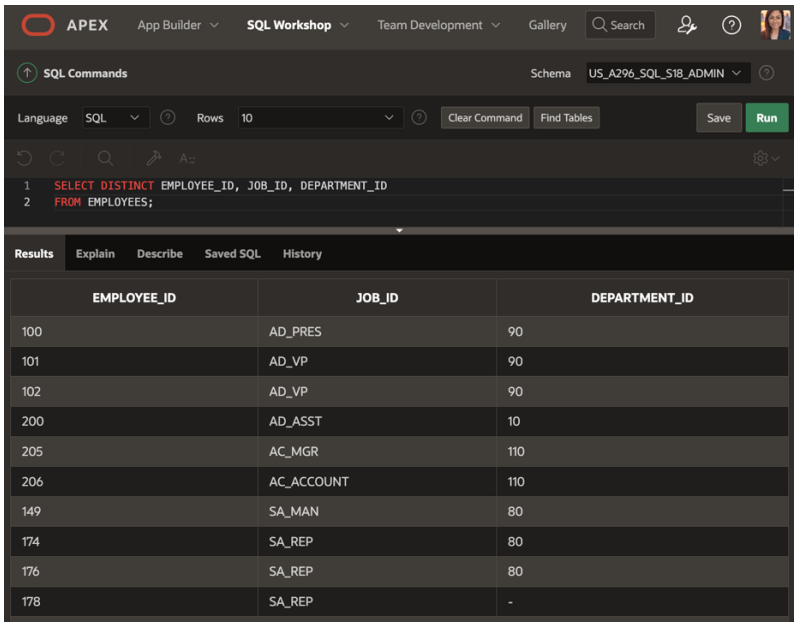


EMPLOYEE_ID	JOB_ID	HIRE_DATE	DEPARTMENT_ID
100	AD_PRES	17-Jun-2002	90
101	AC_MGR	28-Oct-1993	110
101	AD_VP	21-Sep-2004	90
101	AC_ACCOUNT	21-Sep-1989	110
102	AD_VP	13-Jan-2008	90
102	IT_PROG	13-Jan-1993	60
103	IT_PROG	03-Jan-2005	60
104	IT_PROG	21-May-2006	60
107	IT_PROG	07-Feb-2014	60
114	ST_CLERK	24-Mar-1998	50

4. List all employees who have not changed jobs even once. (Such employees are not found in the job_history table)

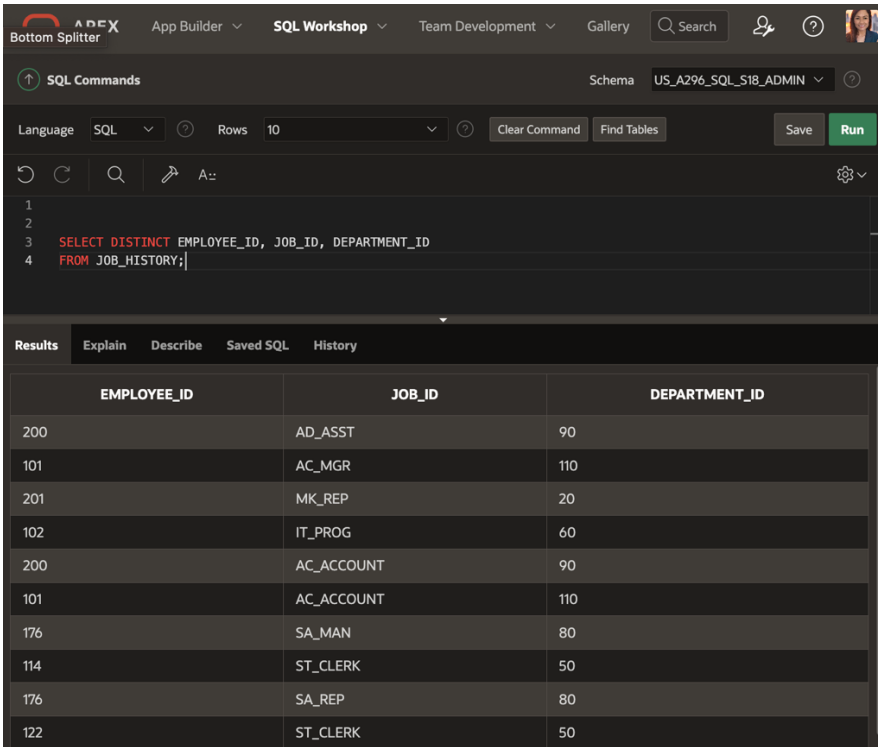
(1) 

(2)



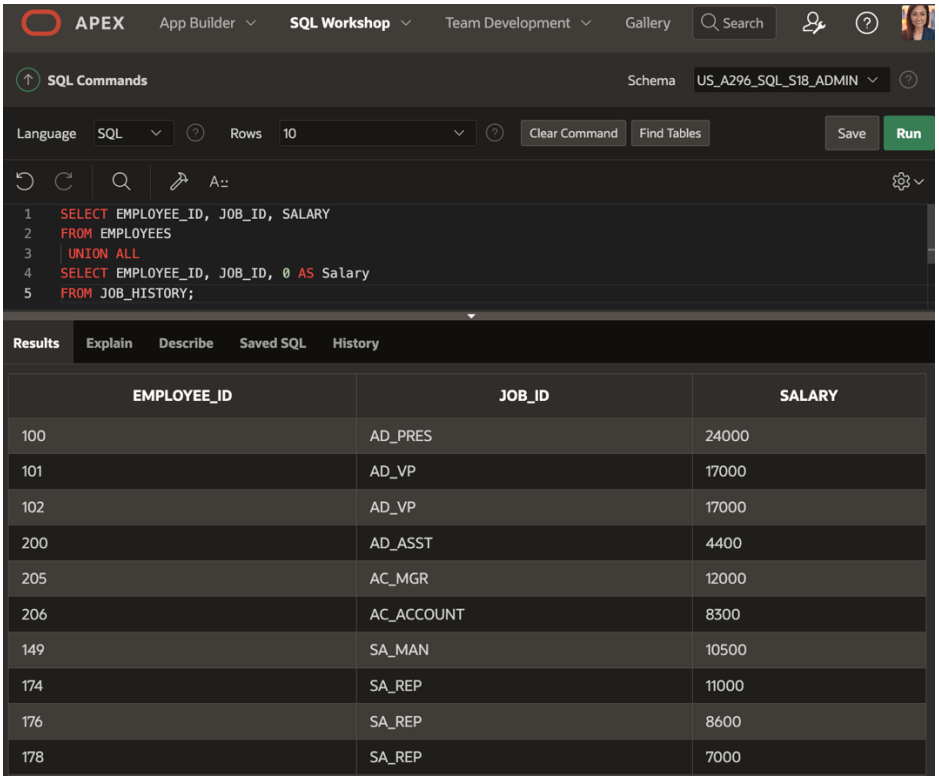
EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID
100	AD_PRES	90
101	AD_VP	90
102	AD_VP	90
200	AD_ASST	10
205	AC_MGR	110
206	AC_ACCOUNT	110
149	SA_MAN	80
174	SA_REP	80
176	SA_REP	80
178	SA_REP	-

5. List the employees that HAVE changed their jobs at least once.



EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID
200	AD_ASST	90
101	AC_MGR	110
201	MK_REP	20
102	IT_PROG	60
200	AC_ACCOUNT	90
101	AC_ACCOUNT	110
176	SA_MAN	80
114	ST_CLERK	50
176	SA_REP	80
122	ST_CLERK	50

6. Using the UNION operator, write a query that displays the employee_id, job_id, and salary of ALL present and past employees. If a salary is not found, then just display a 0 (zero) in its place.



EMPLOYEE_ID	JOB_ID	SALARY
100	AD_PRES	24000
101	AD_VP	17000
102	AD_VP	17000
200	AD_ASST	4400
205	AC_MGR	12000
206	AC_ACCOUNT	8300
149	SA_MAN	10500
174	SA_REP	11000
176	SA_REP	8600
178	SA_REP	7000