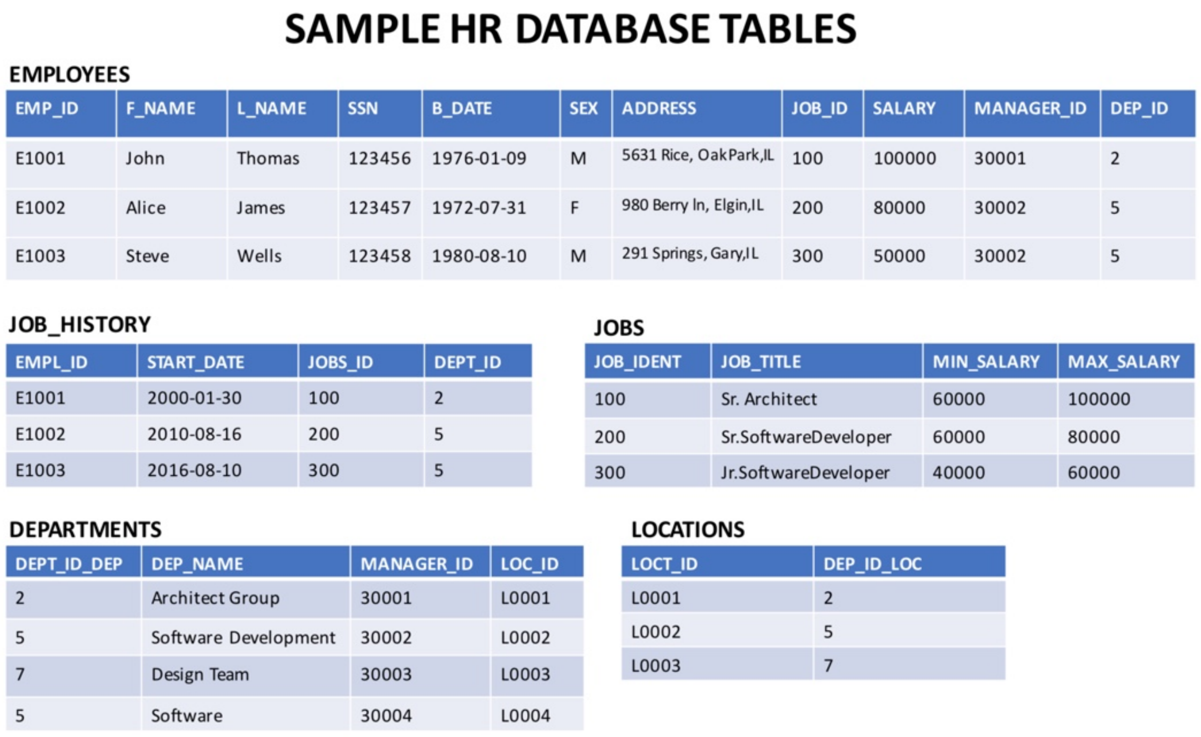
~~LAB: String Patterns, Sorting & Grouping~~

~~The practice problems for this Lab will provide hands on experience with string patterns, sorting result sets and grouping result sets. You will also learn how to run SQL scripts to create several tables at once, as well as how to load data into tables from .csv files.~~

~~HR Database~~

~~We will be working on a sample HR database for this Lab. This HR database schema consists of 5 tables called EMPLOYEES, JOB_HISTORY, JOBS, DEPARTMENTS and LOCATIONS. Each table has a few rows of sample data The following diagram shows the tables for the HR database.~~



SAMPLE HR DATABASE TABLES

EMPLOYEES

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP_ID
E1001	John	Thomas	123456	1976-01-09	M	5631 Rice, OakPark,IL	100	100000	30001	2
E1002	Alice	James	123457	1972-07-31	F	980 Berry ln, Elgin,IL	200	80000	30002	5
E1003	Steve	Wells	123458	1980-08-10	M	291 Springs, Gary,IL	300	50000	30002	5

JOB_HISTORY

EMPL_ID	START_DATE	JOBS_ID	DEPT_ID
E1001	2000-01-30	100	2
E1002	2010-08-16	200	5
E1003	2016-08-10	300	5

JOBS

JOB_IDENT	JOB_TITLE	MIN_SALARY	MAX_SALARY
100	Sr. Architect	60000	100000
200	Sr.SoftwareDeveloper	60000	80000
300	Jr.SoftwareDeveloper	40000	60000

DEPARTMENTS

DEPT_ID_DEP	DEP_NAME	MANAGER_ID	LOC_ID
2	Architect Group	30001	L0001
5	Software Development	30002	L0002
7	Design Team	30003	L0003
5	Software	30004	L0004

LOCATIONS

LOCT_ID	DEP_ID_LOC
L0001	2
L0002	5
L0003	7

~~To complete this lab you will utilize Db2 database service on IBM Cloud as you did for the previous lab. There are three parts to this lab:~~

~~I. Creating tables~~
~~II. Loading data into tables~~
~~III. Composing and running queries~~

~~Part I: CREATING TABLES~~

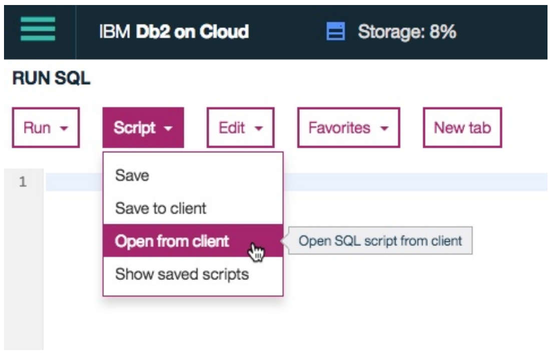
~~If you do not yet have access to Db2 on IBM Cloud, please refer to Lab Instructions in the Module/Week 1.~~

~~Rather than create each table manually by typing the DDL commands in the SQL editor, you will execute a script containing the create table commands for all the tables. Following step by step instructions are provided to perform this:~~

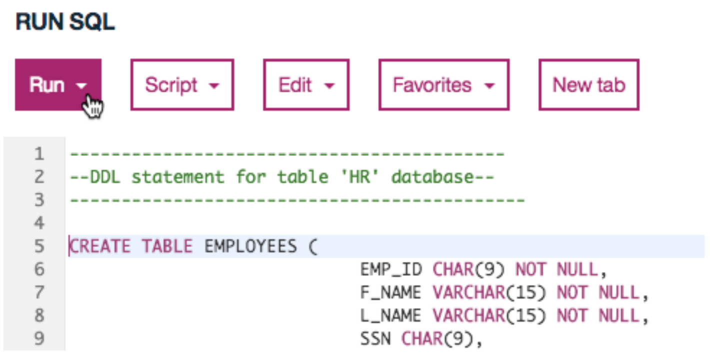
~~1) Download the script file " Script_Create_Tables.sql" provided on the Lab page~~

~~2) Login to IBM Cloud and go to the Dashboard: https://console.bluemix.net/dashboard/apps where you can find the Db2 service that you created in a previous Lab. Click on the service. Next, open the Db2 Console by clicking on "Open Console" button. Go to the Run SQL page. The Run SQL tool enables you to run DDL and SQL statements.~~

~~3) Click on the Folder icon to "Open script". Locate the file Script_Create_Tables.sql that you downloaded to your computer earlier and open it.~~



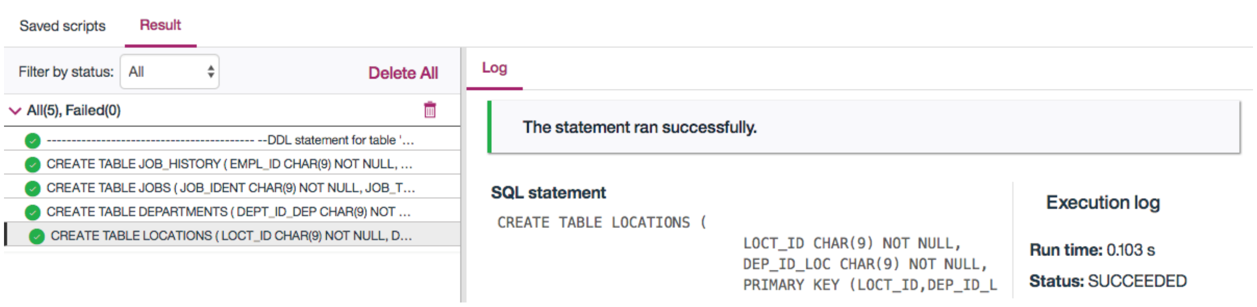
~~4) Once the statements are in the SQL Editor tool , you can run the queries against the database by selecting the option "Run" then "Run All".~~



```
--------------------------------------------
--DDL statement for table 'HR' database--
--------------------------------------------

CREATE TABLE EMPLOYEES (
                            EMP_ID CHAR(9) NOT NULL,
                            F_NAME VARCHAR(15) NOT NULL,
                            L_NAME VARCHAR(15) NOT NULL,
                            SSN CHAR(9),
```

~~5) At the bottom of the screen you will see a Result section. Clicking on a query will show the execution details of the job - whether it ran successfully, or had any errors or warnings. Ensure your queries ran successfully and created all the tables.~~



All(5), Failed(0)

- -- --DDL statement for table '...
- CREATE TABLE JOB_HISTORY (EMPL_ID CHAR(9) NOT NULL, ...
- CREATE TABLE JOBS (JOB_IDENT CHAR(9) NOT NULL, JOB_T...
- CREATE TABLE DEPARTMENTS (DEPT_ID_DEP CHAR(9) NOT ...
- CREATE TABLE LOCATIONS (LOCT_ID CHAR(9) NOT NULL, D...

The statement ran successfully.

SQL statement

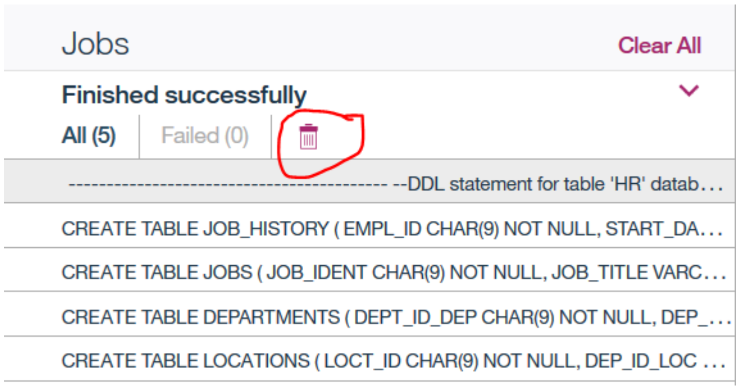
```
CREATE TABLE LOCATIONS (
                                    LOCT_ID CHAR(9) NOT NULL,
                                    DEP_ID_LOC CHAR(9) NOT NULL,
                                    PRIMARY KEY (LOCT_ID,DEP_ID_L
```

Execution log

Run time: 0.103 s

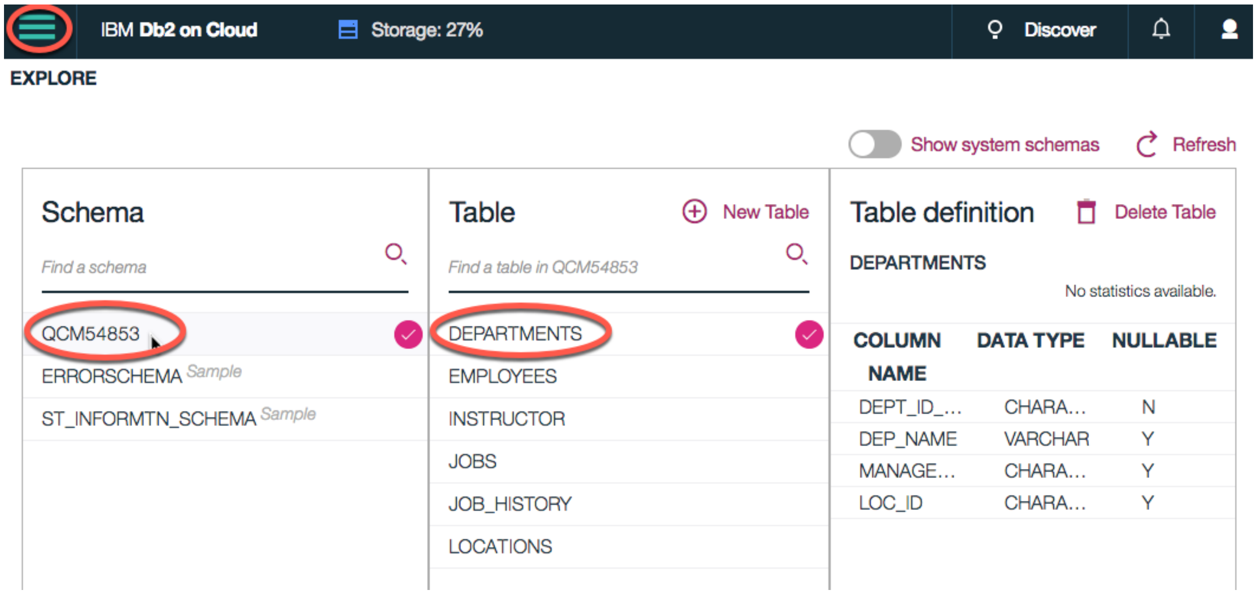
Status: SUCCEEDED

~~6) You can delete the Jobs history by clicking on the Delete icon as shown below.~~



- -- --DDL statement for table 'HR' datab...
- CREATE TABLE JOB_HISTORY (EMPL_ID CHAR(9) NOT NULL, START_DA...
- CREATE TABLE JOBS (JOB_IDENT CHAR(9) NOT NULL, JOB_TITLE VARC...
- CREATE TABLE DEPARTMENTS (DEPT_ID_DEP CHAR(9) NOT NULL, DEP_...
- CREATE TABLE LOCATIONS (LOCT_ID CHAR(9) NOT NULL, DEP_ID_LOC ...

~~7) Now you can look at the tables you created. Navigate to the three bar menu icon and select "Explore". Select the correct Schema to see the newly created tables.~~



Schema: QCM54853, ERRORSCHEMA Sample, ST_INFORMTN_SCHEMA Sample

Table: DEPARTMENTS, EMPLOYEES, INSTRUCTOR, JOBS, JOB_HISTORY, LOCATIONS

DEPARTMENTS

No statistics available.

COLUMN NAME	DATA TYPE	NULLABLE
DEPT_ID_...	CHARA...	N
DEP_NAME	VARCHAR	Y
MANAGE...	CHARA...	Y
LOC_ID	CHARA...	Y

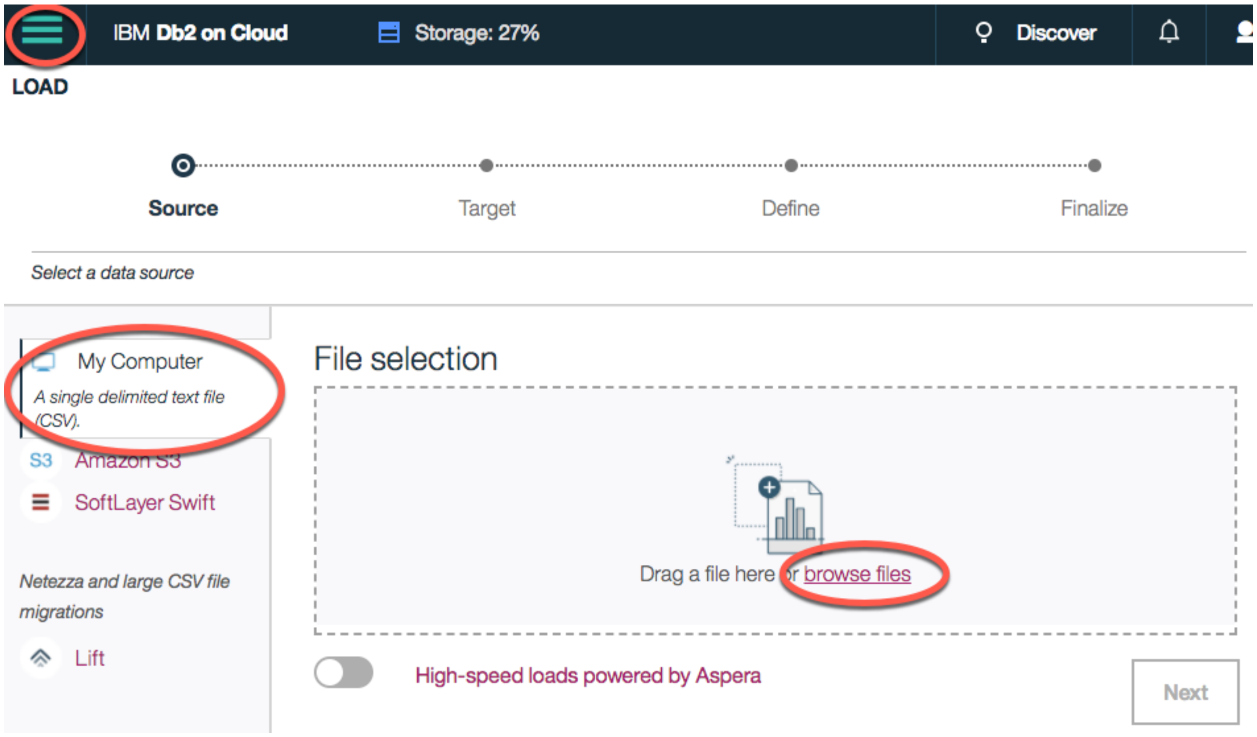
~~Part II: LOADING DATA~~

~~Now let us see how data can be loaded into Db2. We could manually insert each row into the table one by one but that would take a long time. Instead, Db2 (and almost every other database) allows you to Load data from csv files.~~

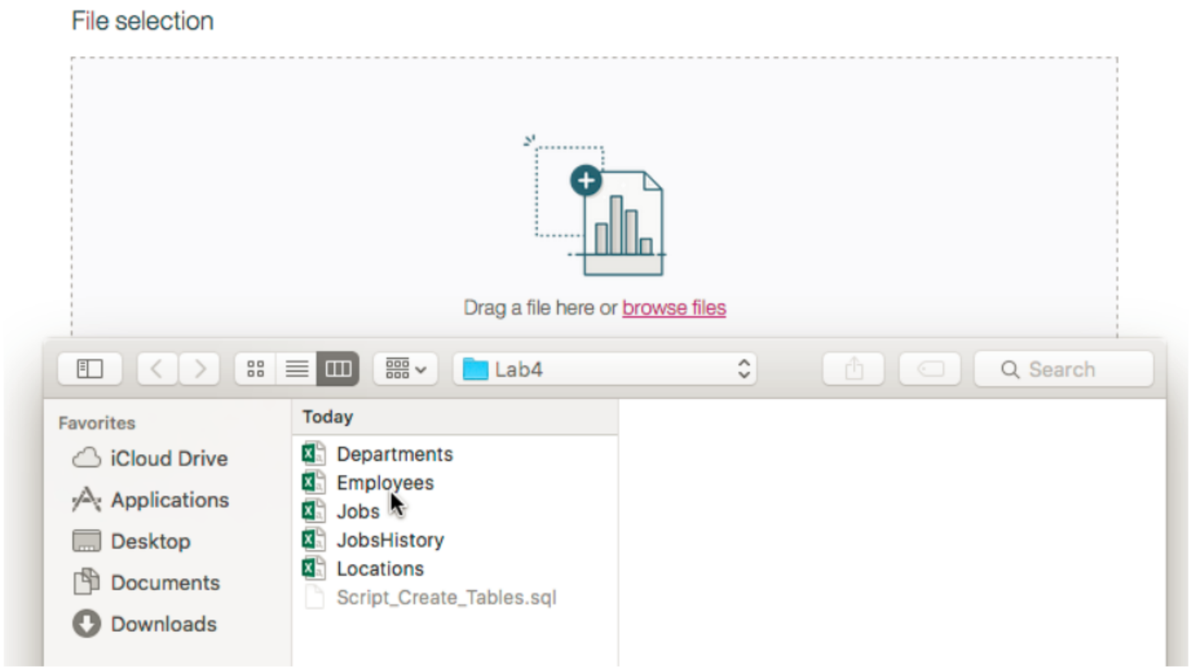
~~Please follow the steps below which explains the process of loading data into the tables we created earlier.~~

~~1) Download the 5 required data source files: ("Employees.csv","Departments.csv","Jobs.csv","JobsHistory.csv","Locations.csv") to your computer:~~

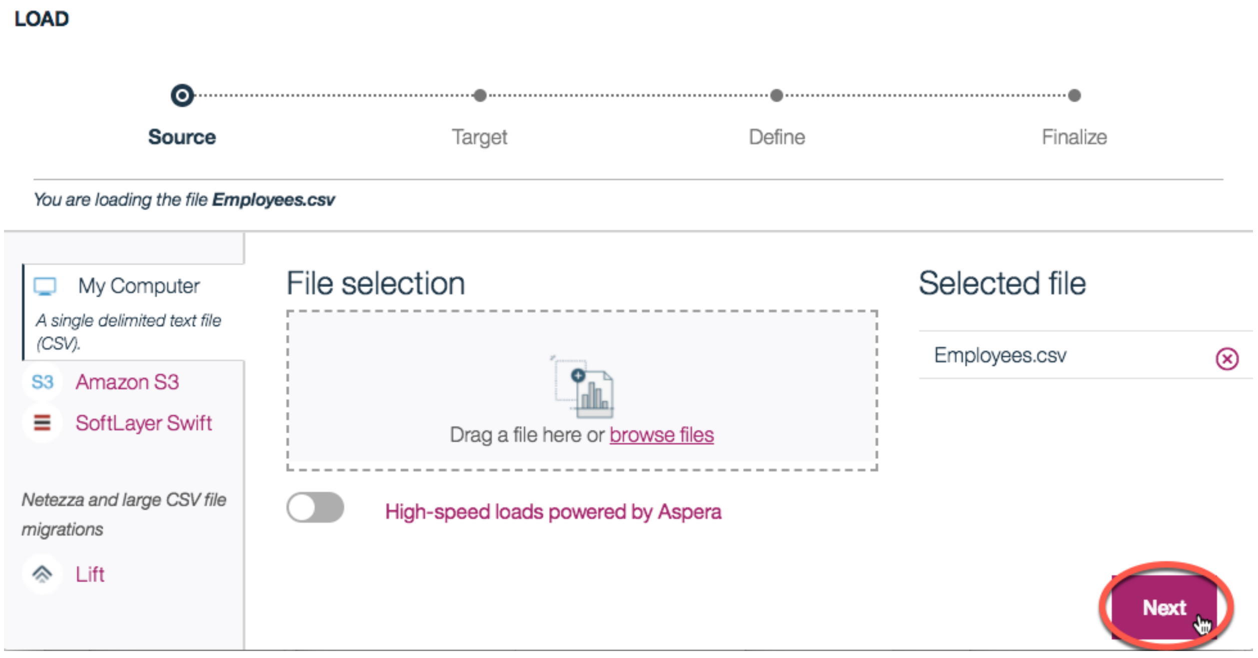
~~2) First let us learn how to load data into the Employees table that we created earlier. From the 3 bar menu icon, navigate to the "Load" page and ensure "My Computer" is selected as the source. Click on the "browse files" link.~~



~~3) Choose the file "Emloyees.csv" that you downloaded to your computer and click "Open".~~

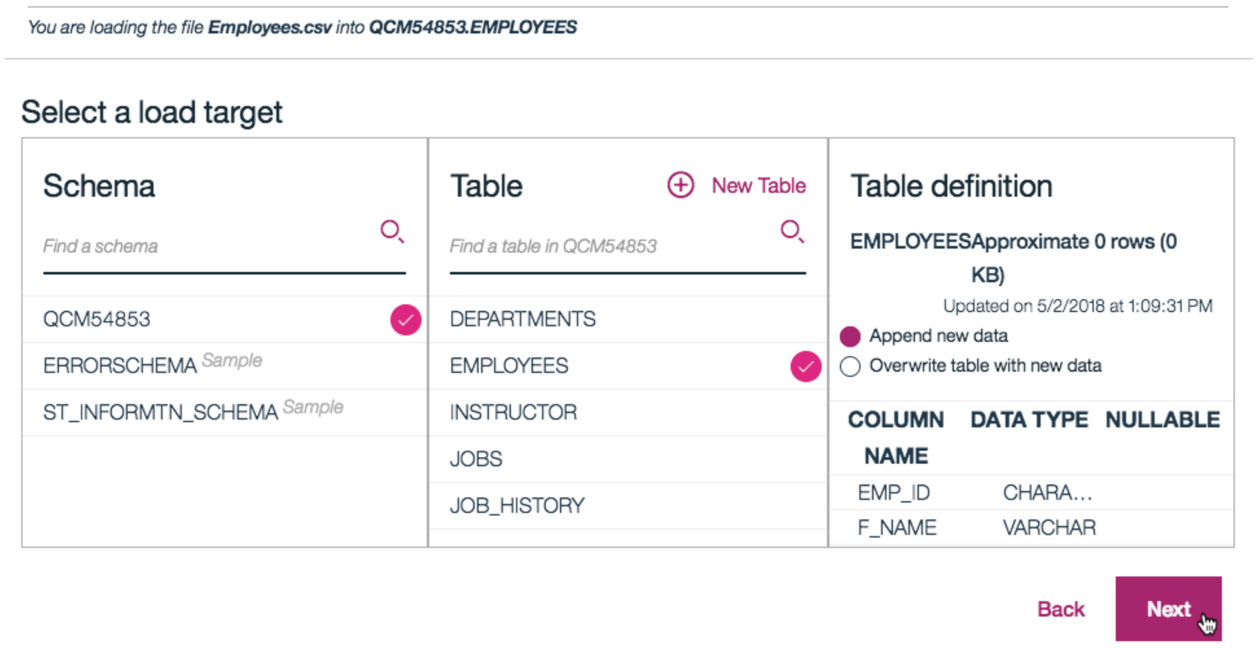


~~4) Once the File is selected click "Next" in the bottom right corner.~~


LOAD
Source
Target
Define
Finalize
My Computer
A single delimited text file (CSV).
Amazon S3
SoftLayer Swift
Netezza and large CSV file migrations
Lift
File selection
Drag a file here or browse files
High-speed loads powered by Aspera
Selected file
Employees.csv
Next

You are loading the file ***Employees.csv***

~~5) Select the schema for your Userid. It will show all the tables that have been created in this Schema previously, including the Employees table. Select the EMPLOYEES table.~~



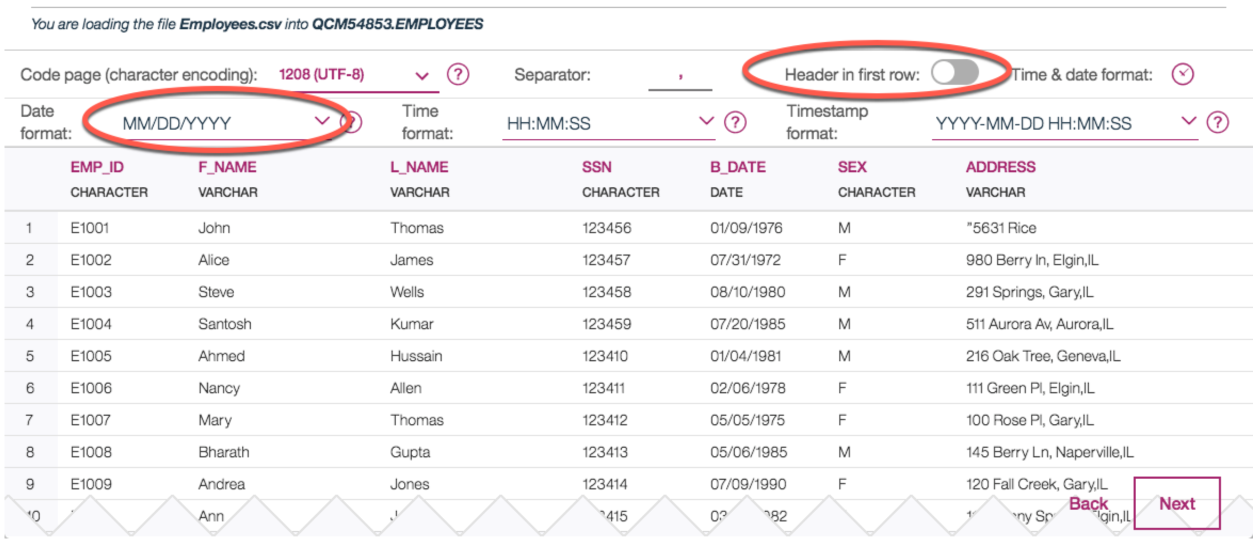
You are loading the file ***Employees.csv*** *into* ***QCM54853.EMPLOYEES***

Select a load target

Schema	Table	Table definition
QCM54853 ✓	DEPARTMENTS	EMPLOYEESApproximate 0 rows (0 KB)
ERRORSCHEMA Sample	EMPLOYEES ✓	Updated on 5/2/2018 at 1:09:31 PM
ST_INFORMTN_SCHEMA Sample	INSTRUCTOR	● Append new data
	JOBS	○ Overwrite table with new data
	JOB_HISTORY	

COLUMN NAME	DATA TYPE	NULLABLE
EMP_ID	CHARA...	
F_NAME	VARCHAR	

~~6) Since our source data files do not contain any rows with column labels, turn off the setting for "Header in first row". Also, click on the down arrow next to "Date format" and choose "MM/DD/YYYY" since that is how the date is formatted in our source file.~~

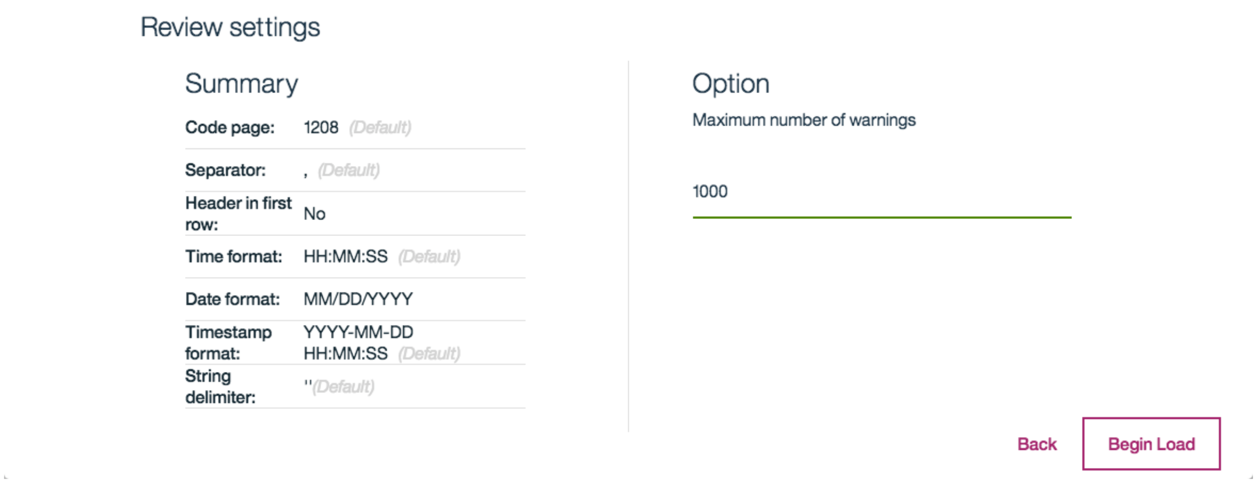


You are loading the file ***Employees.csv*** *into* ***QCM54853.EMPLOYEES***

Code page (character encoding): 1208 (UTF-8)
Separator: ,
Header in first row: off
Time & date format:
Date format: MM/DD/YYYY
Time format: HH:MM:SS
Timestamp format: YYYY-MM-DD HH:MM:SS

	EMP_ID CHARACTER	F_NAME VARCHAR	L_NAME VARCHAR	SSN CHARACTER	B_DATE DATE	SEX CHARACTER	ADDRESS VARCHAR
1	E1001	John	Thomas	123456	01/09/1976	M	"5631 Rice
2	E1002	Alice	James	123457	07/31/1972	F	980 Berry ln, Elgin,IL
3	E1003	Steve	Wells	123458	08/10/1980	M	291 Springs, Gary,IL
4	E1004	Santosh	Kumar	123459	07/20/1985	M	511 Aurora Av, Aurora,IL
5	E1005	Ahmed	Hussain	123410	01/04/1981	M	216 Oak Tree, Geneva,IL
6	E1006	Nancy	Allen	123411	02/06/1978	F	111 Green Pl, Elgin,IL
7	E1007	Mary	Thomas	123412	05/05/1975	F	100 Rose Pl, Gary,IL
8	E1008	Bharath	Gupta	123413	05/06/1985	M	145 Berry Ln, Naperville,IL
9	E1009	Andrea	Jones	123414	07/09/1990	F	120 Fall Creek, Gary,IL
10		Ann		415	0 82		gin,IL

~~7) Click "Next". Review the Load setting and click "Begin Load" in the top-right corner.~~



Review settings

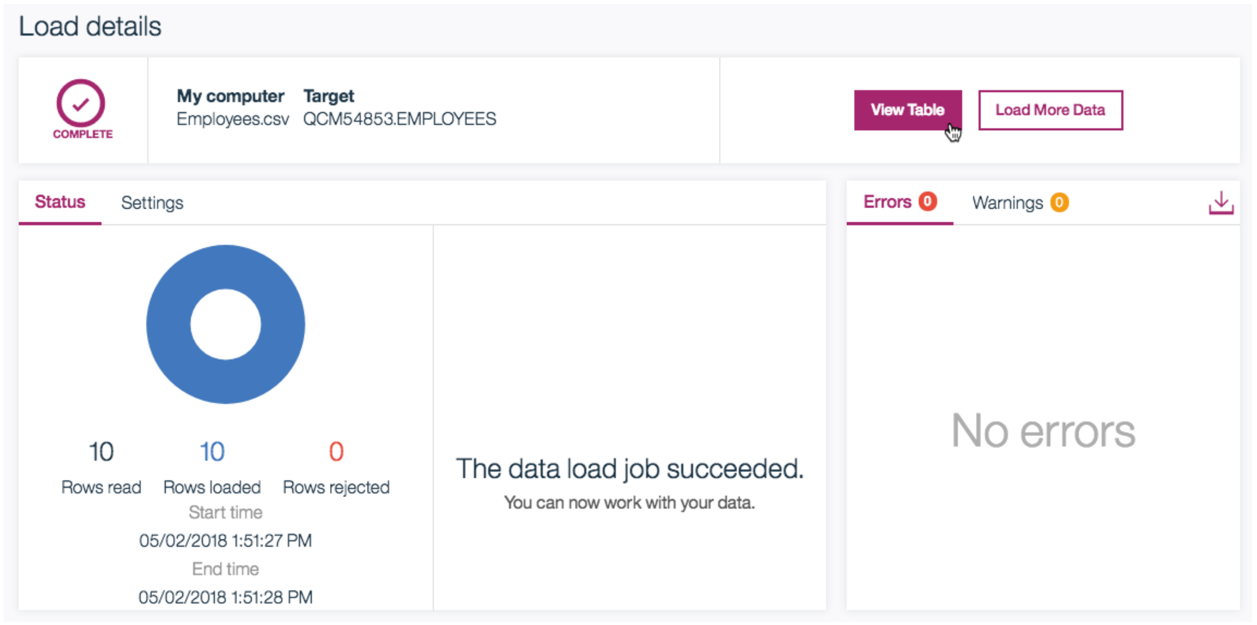
Summary

Code page:	1208 (Default)
Separator:	, (Default)
Header in first row:	No
Time format:	HH:MM:SS (Default)
Date format:	MM/DD/YYYY
Timestamp format:	YYYY-MM-DD HH:MM:SS (Default)
String delimiter:	"(Default)

Option

Maximum number of warnings: 1000

~~8) After Loading is complete you will notice that we were successful in loading all 10 rows of the Employees table. If there are any Errors or Warnings you can view them on this screen.~~



Load details

COMPLETE

My computer: Employees.csv
Target: QCM54853.EMPLOYEES

10 Rows read
10 Rows loaded
0 Rows rejected

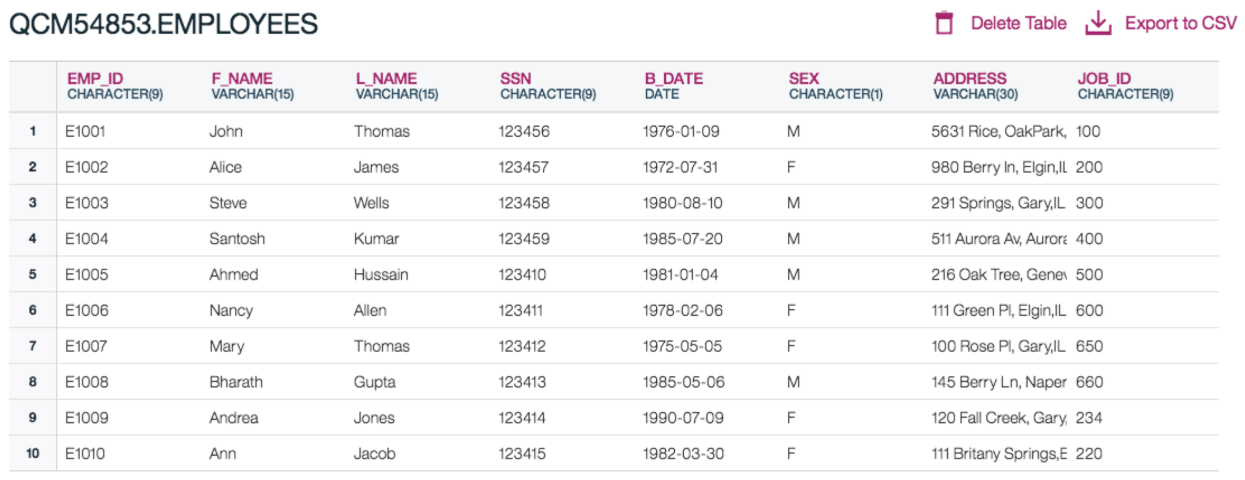
Start time 05/02/2018 1:51:27 PM

End time 05/02/2018 1:51:28 PM

The data load job succeeded.
You can now work with your data.

No errors

~~9) You can see the data that was loaded by clicking on the View Table. Alternatively you can go into the Explore page and page select the correct schema, then the EMPLOYEES table, and click "View Data".~~



QCM54853.EMPLOYEES

Delete Table | Export to CSV

	EMP_ID CHARACTER(9)	F_NAME VARCHAR(15)	L_NAME VARCHAR(15)	SSN CHARACTER(9)	B_DATE DATE	SEX CHARACTER(1)	ADDRESS VARCHAR(30)	JOB_ID CHARACTER(9)
1	E1001	John	Thomas	123456	1976-01-09	M	5631 Rice, OakPark,	100
2	E1002	Alice	James	123457	1972-07-31	F	980 Berry ln, Elgin,IL	200
3	E1003	Steve	Wells	123458	1980-08-10	M	291 Springs, Gary,IL	300
4	E1004	Santosh	Kumar	123459	1985-07-20	M	511 Aurora Av, Aurora	400
5	E1005	Ahmed	Hussain	123410	1981-01-04	M	216 Oak Tree, Genev	500
6	E1006	Nancy	Allen	123411	1978-02-06	F	111 Green Pl, Elgin,IL	600
7	E1007	Mary	Thomas	123412	1975-05-05	F	100 Rose Pl, Gary,IL	650
8	E1008	Bharath	Gupta	123413	1985-05-06	M	145 Berry Ln, Naper	660
9	E1009	Andrea	Jones	123414	1990-07-09	F	120 Fall Creek, Gary,	234
10	E1010	Ann	Jacob	123415	1982-03-30	F	111 Britany Springs,E	220

~~10. Now its your turn to load the remaining 4 tables of the HR database – Locations, JobHistory, Jobs, and Departments. Please follow the steps above to load the data from the remaining source files.~~

~~**Question 1:** Were there any warnings loading data into the JOBS table? What can be done to resolve this?~~

~~Hint: View the data loaded into this table and pay close attention to the JOB_TITLE column.~~

~~**Question 2**: Did all rows from the source file load successfully in the DEPARTMENT table? If not, are you able to figure out why not?~~

~~Hint: Look at the warning. Also, note the Primary Key for this table.~~

~~**Part III: COMPOSING AND RUNNING QUERIES**~~

~~You created the tables for the HR database schema and also learned how to load data into these tables. Now try and work on a few advanced DML queries that were introduced in this module.~~

~~Follow these steps to create and run the queries indicated below~~

~~1) Navigate to the Run SQL tool in Db2 on Cloud~~

~~2) Compose query and run it.~~

~~3) Check the Logs created under the Results section. Looking at the contents of the Log explains whether the SQL statement ran successfully. Also look at the query results to ensure the output is what you expected.~~

~~**Query 1: Retrieve all employees whose address is in Elgin,IL**~~

~~[Hint: Use the LIKE operator to find similar strings]~~

~~**Query 2: Retrieve all employees who were born during the 1970's.**~~

~~[Hint: Use the LIKE operator to find similar strings]~~

~~**Query 3: Retrieve all employees in department 5 whose salary is between 60000 and 70000 .**~~

~~[Hint: Use the keyword BETWEEN for this query]~~

~~**Query 4A: Retrieve a list of employees ordered by department ID.**~~

~~[Hint: Use the ORDER BY clause for this query]~~

~~**Query 4B: Retrieve a list of employees ordered in descending order by department ID and within each department ordered alphabetically in descending order by last name.**~~

~~**Query 5A: For each department ID retrieve the number of employees in the department.**~~

~~[Hint: Use COUNT(*) to retrieve the total count of a column, and then GROUP BY]~~

~~**Query 5B: For each department retrieve the number of employees in the department, and the average employees salary in the department.**~~

~~[Hint: Use COUNT(*) to retrieve the total count of a column, and AVG() function to compute average salaries, and then group]~~

~~**Query 5C: Label the computed columns in the result set of Query 5B as "NUM_EMPLOYEES" and "AVG_SALARY".**~~

~~[Hint: Use **AS "LABEL_NAME"** after the column name]~~

~~**Query 5D: In Query 5C order the result set by Average Salary.**~~

~~[Hint: Use **ORDER BY** after the GROUP BY]~~

~~**Query 5E: In Query 5D limit the result to departments with fewer than 4 employees.**~~

~~[Hint: Use **HAVING** after the GROUP BY, and use the count() function in the HAVING clause instead of the column label.~~

~~Note: WHERE clause is used for filtering the entire result set whereas the HAVING clause is used for filtering the result of the grouping]~~

BONUS Query 6: Similar to 4B but instead of department ID use department name. Retrieve a list of employees ordered by department name, and within each department ordered alphabetically in descending order by last name.

[Hint: Department name is in the DEPARTMENTS table. So your query will need to retrieve data from more than one table. Don't worry if you are not able to figure this one out … we'll cover working with multiple tables in the next lesson.]

In this lab you learned how to work with string patterns, sorting result sets and grouping result sets.

Thank you for completing this lab!

See solutions on the following page.

[redacted]

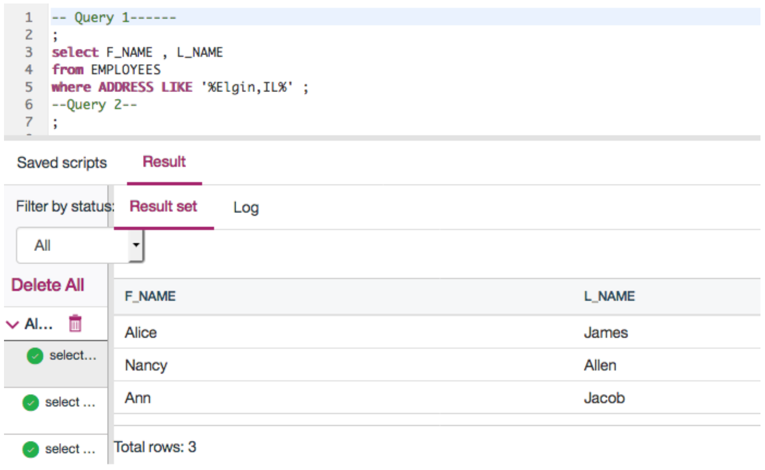
[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

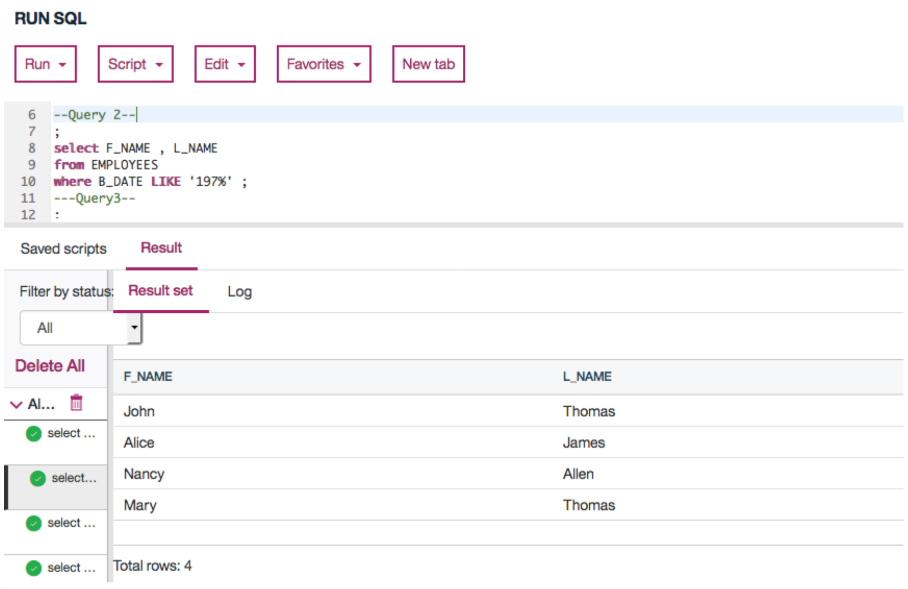


```
-- Query 1------
;
select F_NAME , L_NAME
from EMPLOYEES
where ADDRESS LIKE '%Elgin,IL%' ;
--Query 2--
;
```

F_NAME	L_NAME
Alice	James
Nancy	Allen
Ann	Jacob

Total rows: 3

[redacted]



RUN SQL

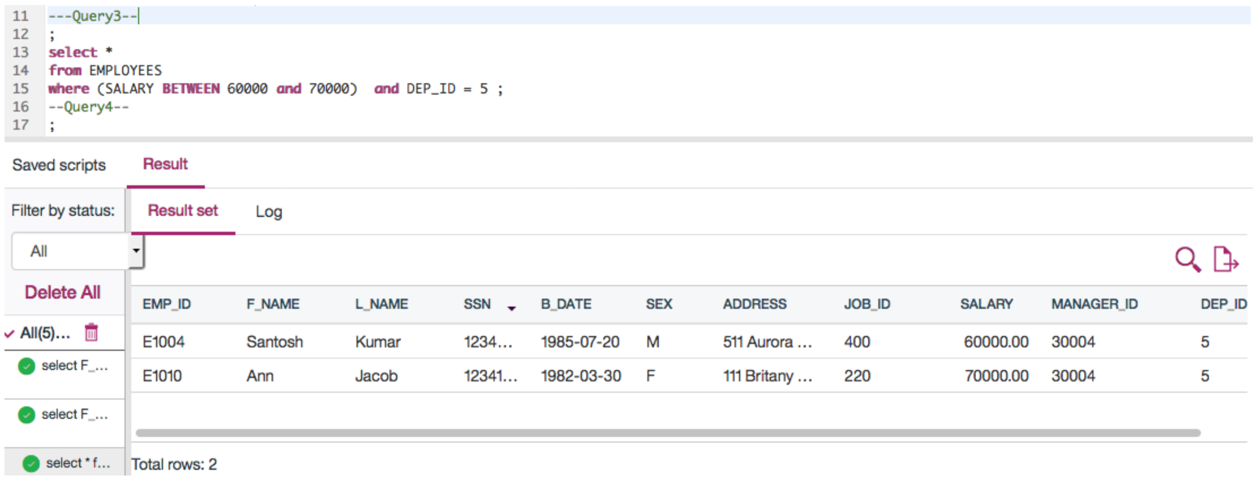
```
--Query 2--
;
select F_NAME , L_NAME
from EMPLOYEES
where B_DATE LIKE '197%' ;
---Query3--
;
```

F_NAME	L_NAME
John	Thomas
Alice	James
Nancy	Allen
Mary	Thomas

Total rows: 4

[redacted]

[redacted]

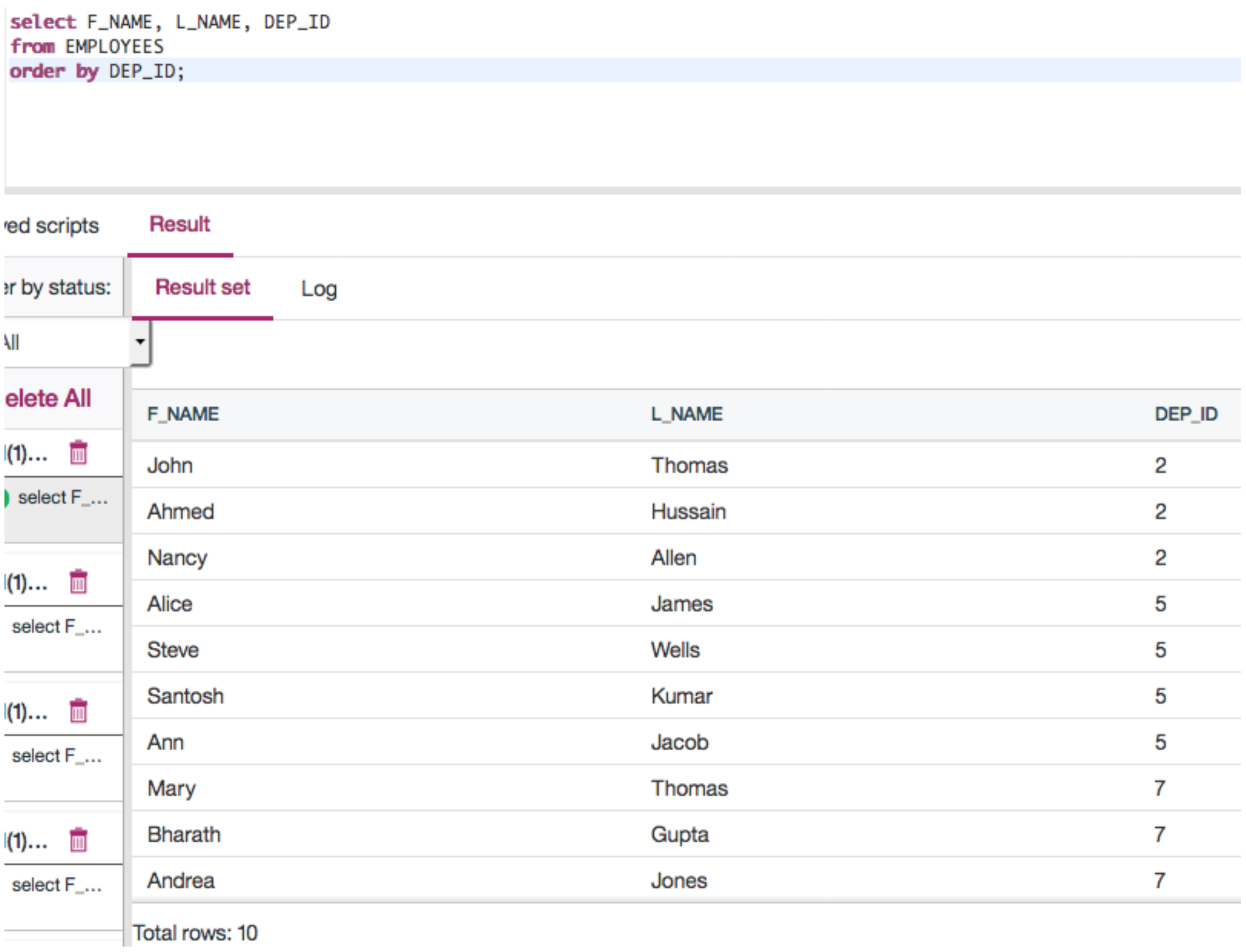


```
---Query3--
;
select *
from EMPLOYEES
where (SALARY BETWEEN 60000 and 70000)  and DEP_ID = 5 ;
--Query4--
;
```

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP_ID
E1004	Santosh	Kumar	1234…	1985-07-20	M	511 Aurora …	400	60000.00	30004	5
E1010	Ann	Jacob	12341…	1982-03-30	F	111 Britany …	220	70000.00	30004	5

Total rows: 2

[redacted]



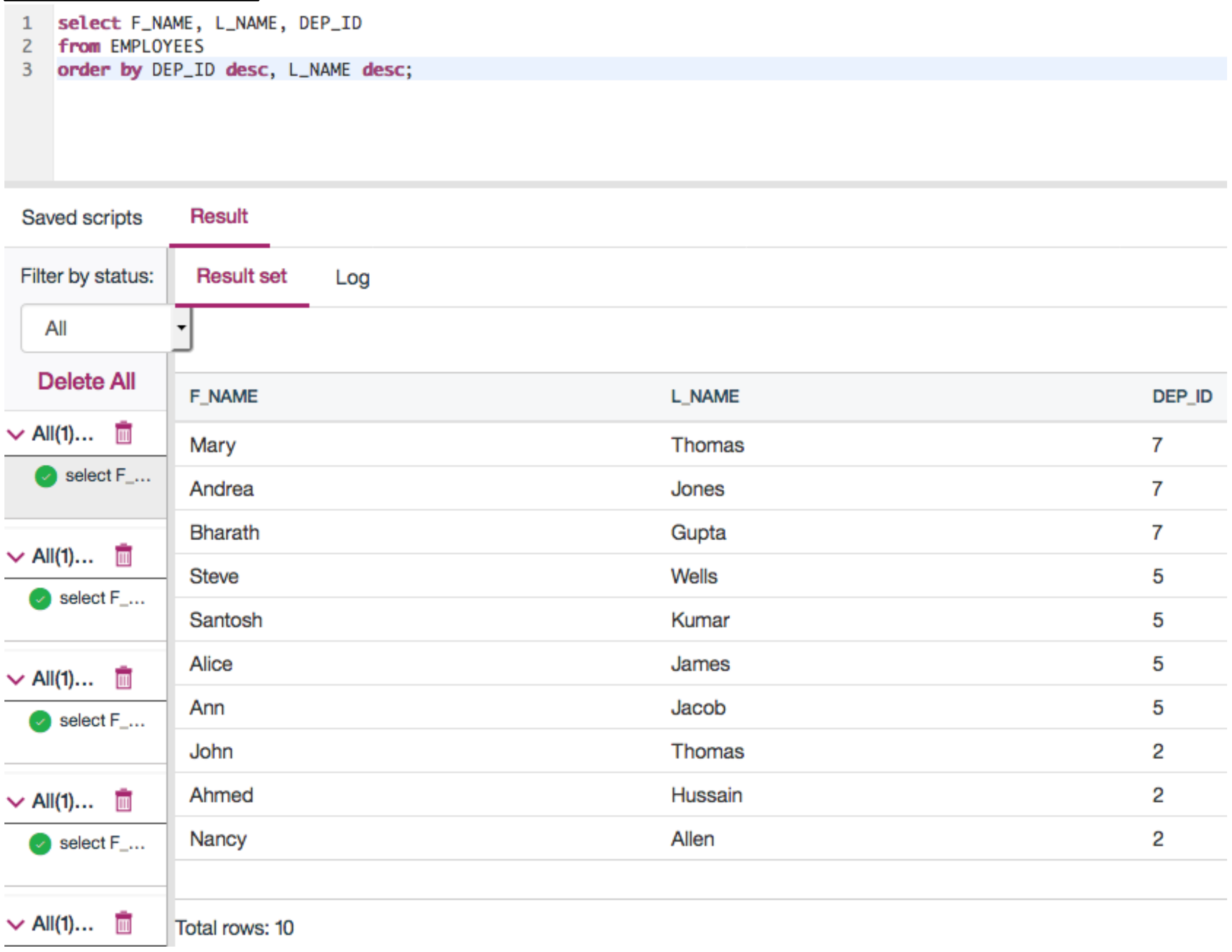
```
select F_NAME, L_NAME, DEP_ID
from EMPLOYEES
order by DEP_ID;
```

F_NAME	L_NAME	DEP_ID
John	Thomas	2
Ahmed	Hussain	2
Nancy	Allen	2
Alice	James	5
Steve	Wells	5
Santosh	Kumar	5
Ann	Jacob	5
Mary	Thomas	7
Bharath	Gupta	7
Andrea	Jones	7

Total rows: 10

[redacted]

[redacted]

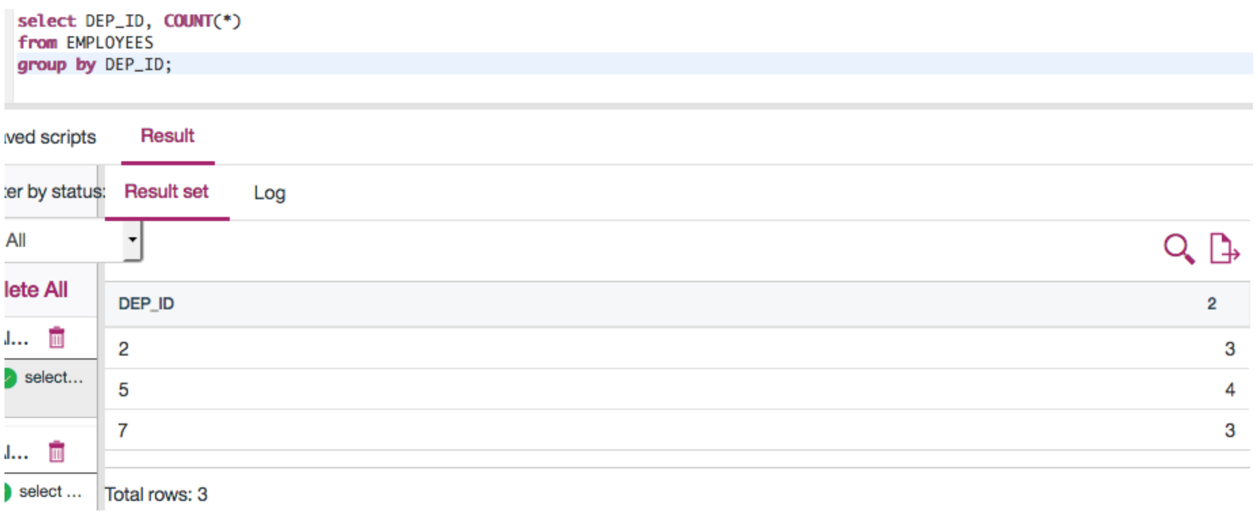


```
select F_NAME, L_NAME, DEP_ID
from EMPLOYEES
order by DEP_ID desc, L_NAME desc;
```

F_NAME	L_NAME	DEP_ID
Mary	Thomas	7
Andrea	Jones	7
Bharath	Gupta	7
Steve	Wells	5
Santosh	Kumar	5
Alice	James	5
Ann	Jacob	5
John	Thomas	2
Ahmed	Hussain	2
Nancy	Allen	2

Total rows: 10

[redacted]



```
select DEP_ID, COUNT(*)
from EMPLOYEES
group by DEP_ID;
```

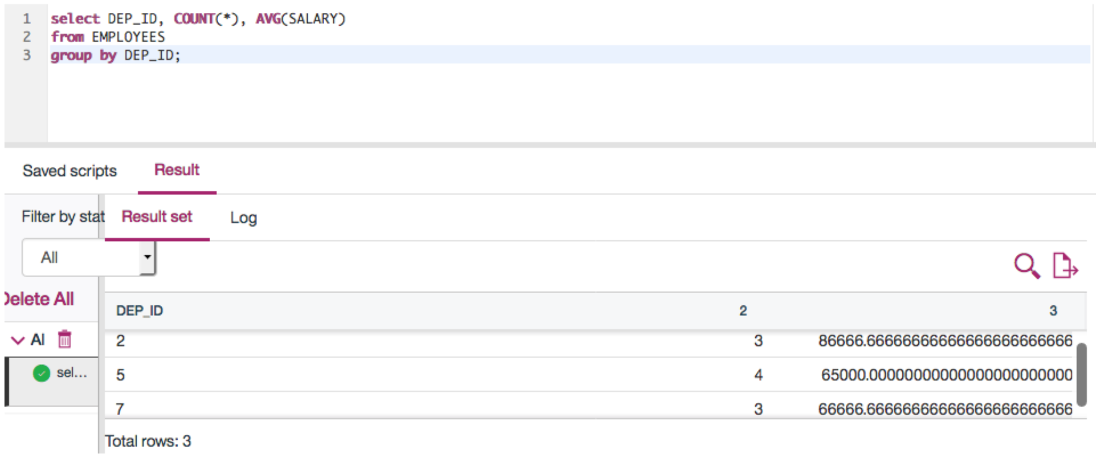
DEP_ID	2
2	3
5	4
7	3

Total rows: 3

[redacted] [redacted]

[redacted]

[redacted]

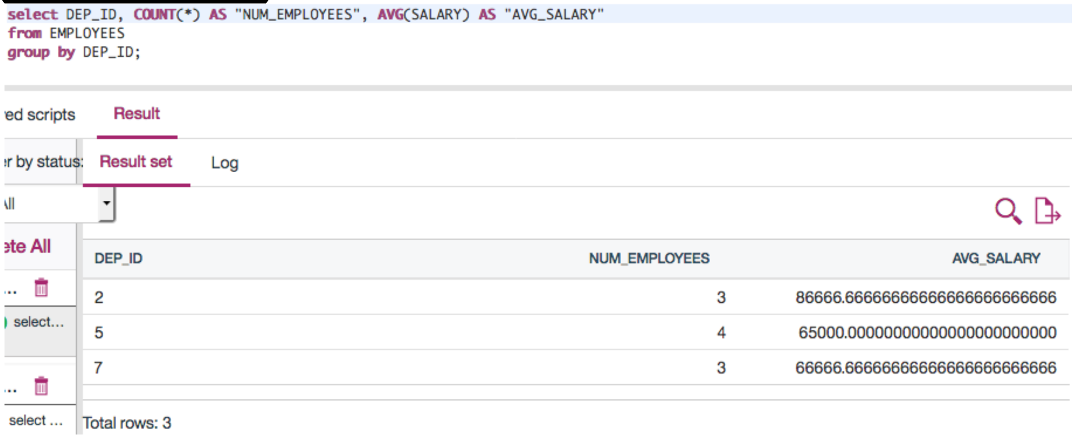


```
select DEP_ID, COUNT(*), AVG(SALARY)
from EMPLOYEES
group by DEP_ID;
```

DEP_ID	2	3
2	3	86666.66666666666666666666666
5	4	65000.00000000000000000000000
7	3	66666.66666666666666666666666

Total rows: 3

[redacted]

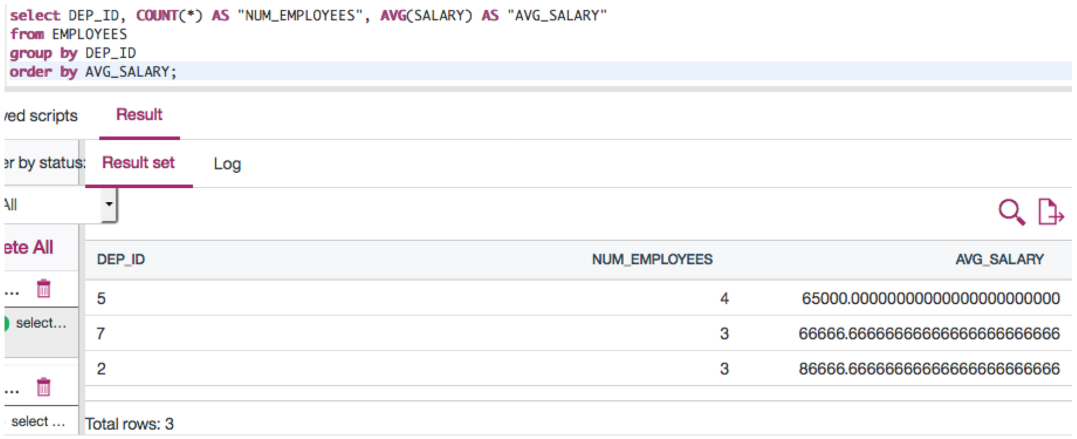


```
select DEP_ID, COUNT(*) AS "NUM_EMPLOYEES", AVG(SALARY) AS "AVG_SALARY"
from EMPLOYEES
group by DEP_ID;
```

DEP_ID	NUM_EMPLOYEES	AVG_SALARY
2	3	86666.66666666666666666666666
5	4	65000.00000000000000000000000
7	3	66666.66666666666666666666666

Total rows: 3

[redacted]



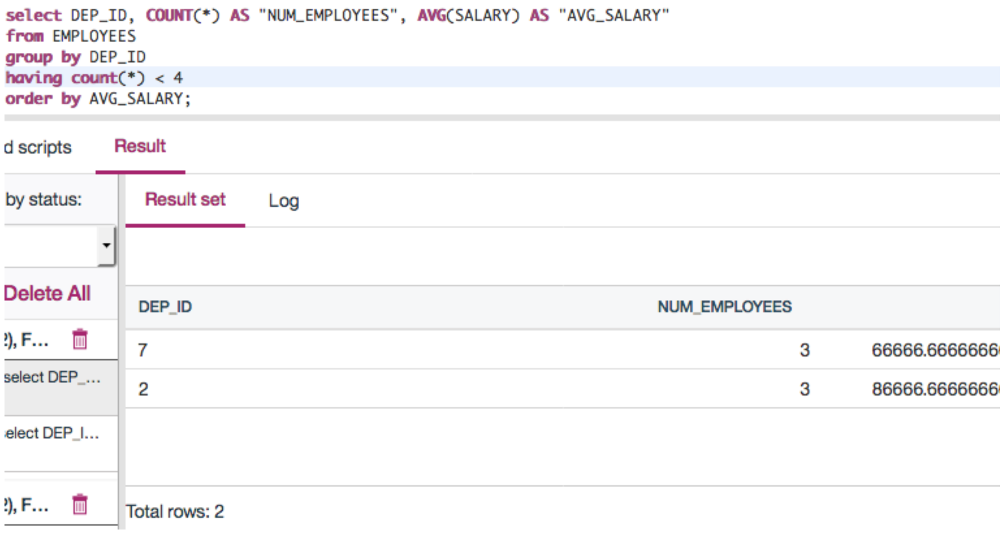
```
select DEP_ID, COUNT(*) AS "NUM_EMPLOYEES", AVG(SALARY) AS "AVG_SALARY"
from EMPLOYEES
group by DEP_ID
order by AVG_SALARY;
```

DEP_ID	NUM_EMPLOYEES	AVG_SALARY
5	4	65000.00000000000000000000000
7	3	66666.66666666666666666666666
2	3	86666.66666666666666666666666

Total rows: 3

[redacted]

Query 5E: Output



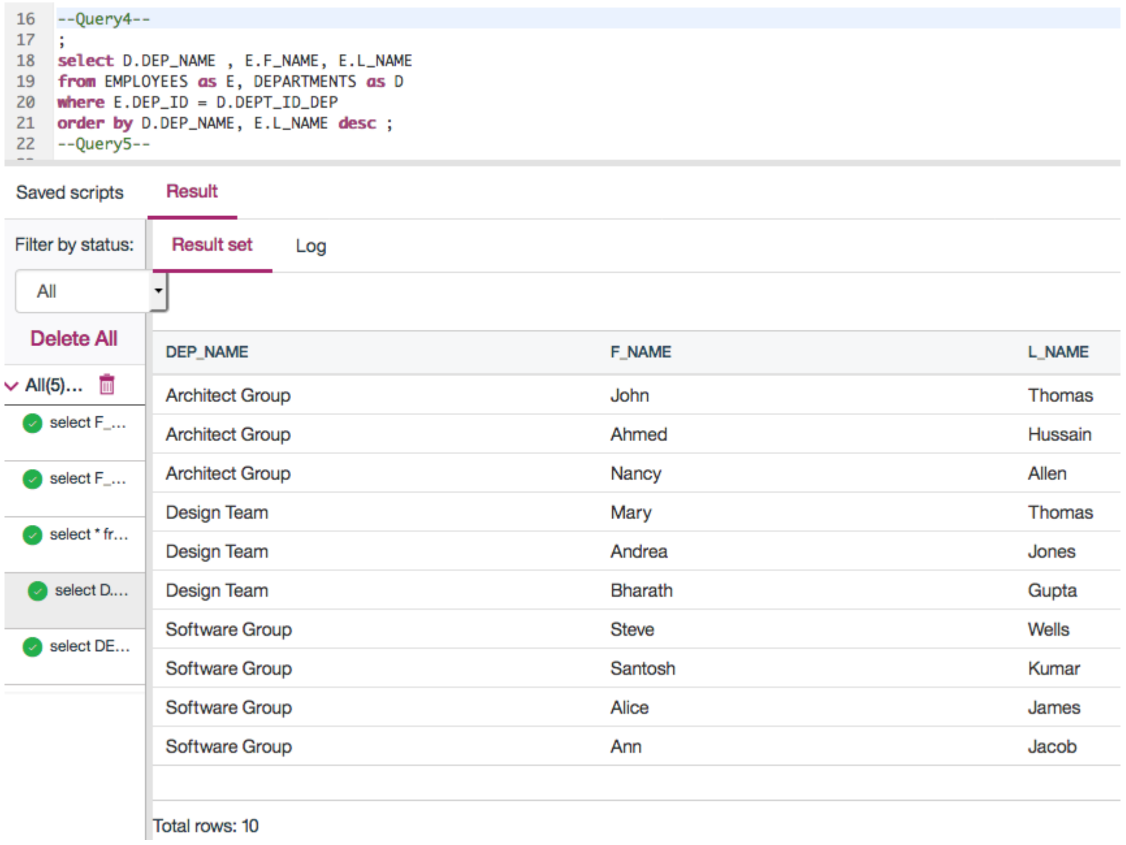
```
select DEP_ID, COUNT(*) AS "NUM_EMPLOYEES", AVG(SALARY) AS "AVG_SALARY"
from EMPLOYEES
group by DEP_ID
having count(*) < 4
order by AVG_SALARY;
```

DEP_ID	NUM_EMPLOYEES	
7	3	66666.6666666
2	3	86666.6666666

Total rows: 2

BONUS Query 6: Output

Note that in the Query below "D" and "E" are aliases for the table names. Once you define an alias like "D' in your query, you can simply write "D.COLUMN_NAME" rather than the full form 'DEPARTMENTS.COLUMN_NAME".



```
--Query4--
;
select D.DEP_NAME , E.F_NAME, E.L_NAME
from EMPLOYEES as E, DEPARTMENTS as D
where E.DEP_ID = D.DEPT_ID_DEP
order by D.DEP_NAME, E.L_NAME desc ;
--Query5--
```

DEP_NAME	F_NAME	L_NAME
Architect Group	John	Thomas
Architect Group	Ahmed	Hussain
Architect Group	Nancy	Allen
Design Team	Mary	Thomas
Design Team	Andrea	Jones
Design Team	Bharath	Gupta
Software Group	Steve	Wells
Software Group	Santosh	Kumar
Software Group	Alice	James
Software Group	Ann	Jacob

Total rows: 10

[redacted] [redacted]