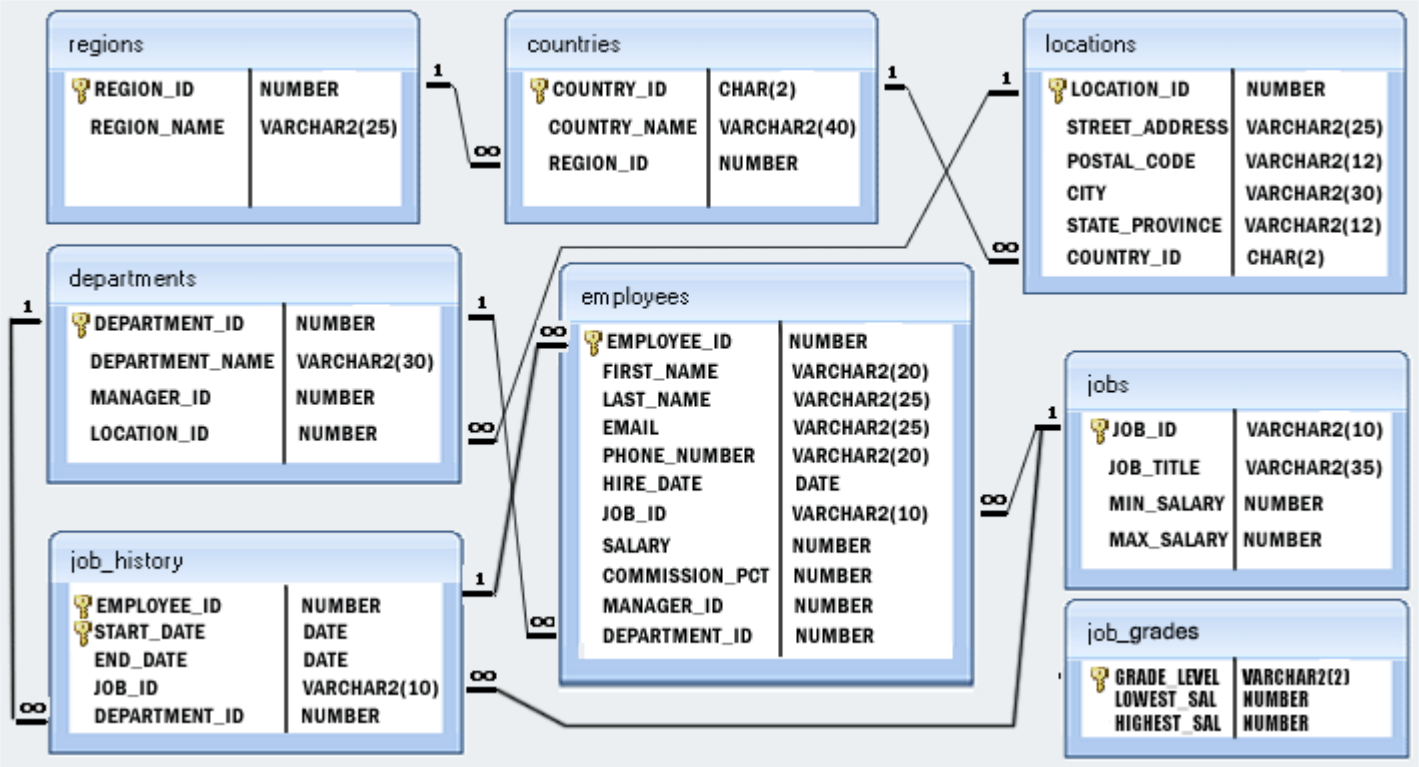
INSERT Statement Examples

The Human Resources (HR) Database

This sample database consists of 8 tables, as depicted in the following Entity-Relationship (ER) Diagram.



Run the SQL Scripts

Before we proceed any further, please go ahead and import the SQL Script (click here (https://bryanuniversity.instructure.com/courses/10152/files/744673/download?wrap=1) (https://bryanuniversity.instructure.com/courses/10152/files/744673/download?wrap=1) to download the script file) into MySQL Work Bench and/or PostgreSQL applications.

Insert Statements

```
mysql> SELECT * FROM countries;
+------------+--------------+-----------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+-----------+
| C1         | India        |      NULL |
+------------+--------------+-----------+
1 row in set (0.00 sec)
```

- Write a SQL statement to create duplicate of countries table named country_new with all structure and data.

```
CREATE TABLE IF NOT EXISTS country_new
AS SELECT * FROM countries;
```

Sample Output:

```
mysql> SHOW COLUMNS FROM country_new;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| COUNTRY_ID   | varchar(8)    | YES  |     | NULL    |       |
| COUNTRY_NAME | varchar(40)   | YES  |     | NULL    |       |
| REGION_ID    | decimal(10,0) | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM country_new;
+------------+--------------+-----------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+-----------+
| C1         | India        |      1001 |
+------------+--------------+-----------+
1 row in set (0.00 sec)
```

- Write a SQL statement to insert NULL values against region_id column for a row of countries table.

```
INSERT INTO countries (country_id,country_name,region_id) VALUES('C1','India',NULL);
```

Sample Output:

```
mysql> SELECT * FROM countries;
+------------+--------------+-----------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+-----------+
| C1         | India        |      NULL |
+------------+--------------+-----------+
1 row in set (0.00 sec)
```

- Write a SQL statement to insert 3 rows by a single insert statement.

```
INSERT INTO countries VALUES
('C0001','India',1001),
('C0002','USA',1007),
('C0003','UK',1003);
```

Sample Output:

```
mysql> SELECT * FROM COUNTRIES;
+------------+--------------+-----------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+-----------+
| C0001      | India        |      1001 |
| C0002      | USA          |      1007 |
| C0003      | UK           |      1003 |
+------------+--------------+-----------+
3 rows in set (0.00 sec)
```

- Write a SQL statement insert rows from country_new table to countries table.

```
INSERT INTO countries
SELECT * FROM country_new;
```

Sample Output:

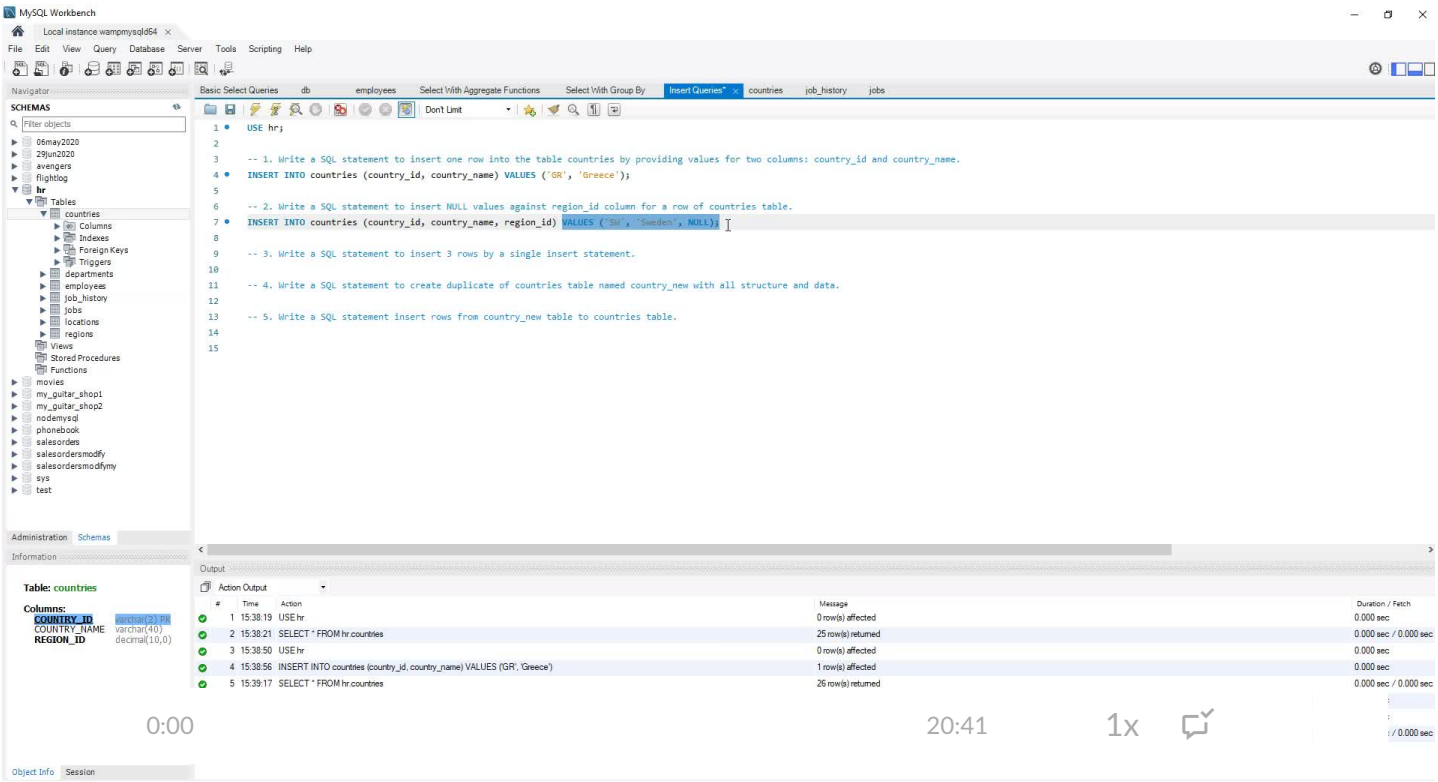
```
mysql> SELECT * FROM country_new;
+------------+--------------+-----------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+-----------+
| C0001      | India        |      1001 |
| C0002      | USA          |      1007 |
| C0003      | UK           |      1003 |
+------------+--------------+-----------+
3 rows in set (0.00 sec)
```

- Write a SQL statement to insert one row in jobs table to ensure that no duplicate value will be entered in the job_id column.

```
CREATE TABLE IF NOT EXISTS jobs (
JOB_ID integer NOT NULL UNIQUE ,
JOB_TITLE varchar(35) NOT NULL,
MIN_SALARY decimal(6,0)
);

INSERT INTO jobs VALUES(1001,'OFFICER',8000);
```

Sample Output:



- Write a SQL statement to insert a record with your own value into the table countries against each columns.

```
INSERT INTO countries VALUES('C1','India',1001);
```

Sample Output:

```
mysql> SELECT * FROM countries;
+------------+--------------+------------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+------------+
| C1         | India        |       1001 |
+------------+--------------+------------+
1 row in set (0.00 sec)
```

- Write a SQL statement to insert one row into the table countries against the column country_id and country_name.

```
INSERT INTO countries (country_id,country_name) VALUES('C1','India');
```

Sample Output:

```
mysql> INSERT INTO jobs VALUES(1001,'OFFICER',8000);
ERROR 1062 (23000): Duplicate entry '1001' for key 'JOB_ID'
```

- Write a SQL statement to insert a record into the table countries to ensure that, a country_id and region_id combination will be entered once in the table.

```
INSERT INTO countries VALUES(501,'Italy',185);
```

Sample Output:

```
mysql> INSERT INTO countries VALUES(501,'Italy',185);
ERROR 1062 (23000): Duplicate entry '501-185' for key 'PRIMARY'
```

- Write a SQL statement to insert rows into the table countries in which the value of country_id column will be unique and auto incremented.

```
INSERT INTO countries(COUNTRY_NAME,REGION_ID) VALUES('India',185);
```

Sample Output:

```
mysql> SELECT * FROM countries;
+------------+--------------+-----------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+-----------+
|          1 | India        |       185 |
+------------+--------------+-----------+
1 row in set (0.00 sec)
```

```
INSERT INTO countries(COUNTRY_NAME,REGION_ID) VALUES('Japan',102);
```

```
mysql> SELECT * FROM countries;
+------------+--------------+-----------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+-----------+
|          1 | India        |       185 |
|          2 | Japan        |       102 |
+------------+--------------+-----------+
2 rows in set (0.03 sec)
```

- Write a SQL statement to insert records into the table countries to ensure that the country_id column will not contain any duplicate data and this will be automatically incremented and the column country_name will be filled up by 'N/A' if no value assigned for that column.

```
INSERT INTO countries VALUES(501,'India',102);
```

Sample Output:

```
mysql> SELECT * FROM countries;
+------------+--------------+-----------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+-----------+
|        501 | India        |       102 |
+------------+--------------+-----------+
1 row in set (0.00 sec)
```

```
INSERT INTO countries(region_id) VALUES(109);
```

```
mysql> SELECT * FROM countries;
+------------+--------------+-----------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+-----------+
|        501 | India        |       102 |
|        502 | N/A          |       109 |
+------------+--------------+-----------+
2 rows in set (0.00 sec)
```

```
INSERT INTO countries(country_name,region_id) VALUES('Australia',121);
```

```
mysql> SELECT * FROM countries;
+------------+--------------+-----------+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+------------+--------------+-----------+
|        501 | India        |       102 |
|        502 | N/A          |       109 |
|        503 | Australia    |       121 |
+------------+--------------+-----------+
3 rows in set (0.00 sec)
```

- Write a SQL statement to insert rows in the job_history table in which one column job_id is containing those values which are exists in job_id column of jobs table.

```
INSERT INTO job_history VALUES(501,1001,60);
```

```
mysql> SELECT * FROM job_history;
+-------------+--------+---------------+
| EMPLOYEE_ID | JOB_ID | DEPARTMENT_ID |
+-------------+--------+---------------+
|         501 |   1001 |            60 |
+-------------+--------+---------------+
1 row in set (0.00 sec)
```

The value against job_id is 1001 which is exists in the job_id column of the jobs table, so no problem arises.

Now insert another row in the job_history table.

```
INSERT INTO job_history VALUES(502,1003,80);
```

```
mysql> INSERT INTO job_history VALUES(502,1003,80);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`hrr`.`job_history
 (`JOB_ID`) REFERENCES `jobs` (`JOB_ID`))
```

Here in the above, the value against job_id is 1003 which is not exists in the job_id column of the jobs(parent table) table and that is why the child table job_history cannot contain the value of job_id as specified. Here the primary key - foreign key relationship is violating and shows the above message.

- Write a SQL statement to insert rows into the table employees in which a set of columns department_id and manager_id contains a unique value and that combined values must have exists into the table departments.

```
INSERT INTO employees VALUES(510,'Alex','Hanes','CLERK',18000,201,60);

INSERT INTO employees VALUES(511,'Kim','Leon','CLERK',18000,211,80);
```

Sample Output:

```
mysql> SELECT * FROM employees;
+-------------+------------+-----------+--------+----------+------------+-------
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | JOB_ID | SALARY   | MANAGER_ID | DEPART
+-------------+------------+-----------+--------+----------+------------+-------
|         510 | Alex       | Hanes     | CLERK  | 18000.00 |        201 |
|         511 | Kim        | Leon      | CLERK  | 18000.00 |        211 |
+-------------+------------+-----------+--------+----------+------------+-------
2 rows in set (0.00 sec)
```

The value against department_id and manager_id combination (60,201) and (80,211) are unique in the departmentis(parent) table so, there is no problem arise to insert the rows in the child table employees.

Now insert another row in the employees table.

```
INSERT INTO employees VALUES(512,'Kim','Leon','CLERK',18000,80,211);
```

```
mysql> INSERT INTO employees VALUES(512,'Kim','Leon','CLERK',18000,80,211);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`hrr`.`employees`, CONSTRAI
EPARTMENT_ID`, `MANAGER_ID`) REFERENCES `departments` (`DEPARTMENT_ID`, `MANAGER_ID`))
```

Here in the above, the value against department_id and manager_id combination (211,80) does not matching with the same combination in departments(parent table) table and that is why the child table employees can not contain the combination of values including department_id and manager_id

as specified. Here the primary key - foreign key relationship is being violated and shows the above message.

- Write a SQL statement to insert rows into the table employees in which a set of columns department_id and job_id contains the values which must have exists into the table departments and jobs.

```
INSERT INTO employees VALUES(510,'Alex','Hanes',60,1001,18000);
```

Sample Output:

```
mysql> SELECT * FROM employees;
+-------------+------------+-----------+---------------+--------+----------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | DEPARTMENT_ID | JOB_ID | SALARY   |
+-------------+------------+-----------+---------------+--------+----------+
|         510 | Alex       | Hanes     |            60 |   1001 | 18000.00 |
+-------------+------------+-----------+---------------+--------+----------+
1 row in set (0.00 sec)
```

Here in the above insert statement the child column department_id and job_id of child table employees are successfully referencing with the department_id and job_id column of parent tables departments and jobs respectively, so no problem have been arisen to the insertion.

Now insert another row in the employees table.

```
INSERT INTO employees VALUES(511,'Tom','Elan',60,1003,22000);
```

```
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`hrr`.`employees`,
OB_ID`) REFERENCES `jobs` (`JOB_ID`))
```

Here in the above insert statement show that, within child columns department_id and job_id of child table employees, the department_id are successfully referencing with the department_id of parent table departments but job_id column are not successfully referencing with the job_id of parent table jobs, so the problem have been arisen to the insertion displayed an error message.

Now insert another row in the employees table.

```
INSERT INTO employees VALUES(511,'Tom','Elan',80,1001,22000);
```

```
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`hrr`.`employees`, CONSTRAI
OB_ID`) REFERENCES `jobs` (`JOB_ID`))
```

Here in the above insert statement show that, within child columns department_id and job_id of child table employees, the job_id are successfully referencing with the job_id of parent table jobs but department_id column are not successfully referencing with the department_id of parent table departments, so the problem have been arisen to the insertion and displayed the error message.