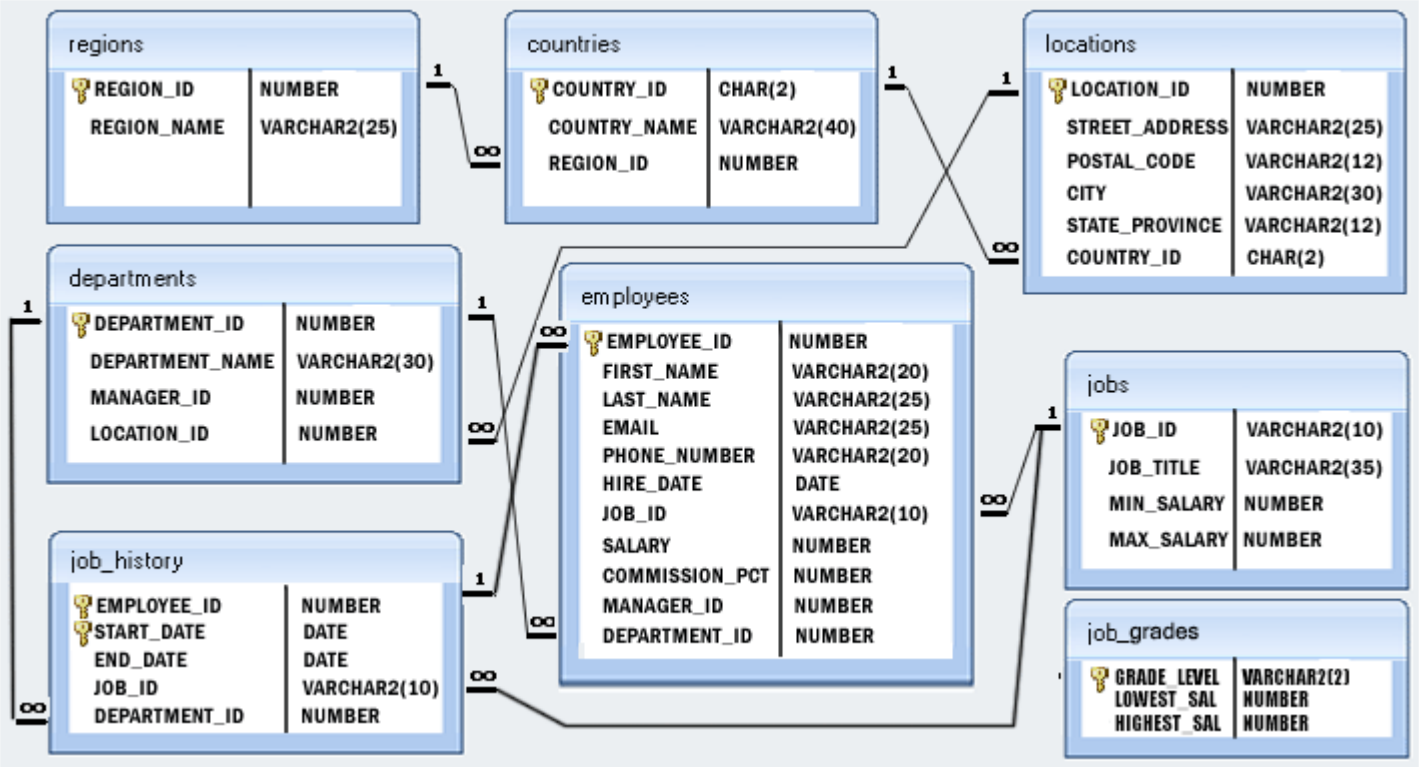
ALTER Statement Examples

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.

Alter Statements

```
mysql> SHOW COLUMNS FROM locations;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| LOCATION_ID    | decimal(4,0) | YES  |     | NULL    |       |
| STREET_ADDRESS | varchar(40)  | YES  |     | NULL    |       |
| POSTAL_CODE    | varchar(12)  | YES  |     | NULL    |       |
| CITY           | varchar(30)  | YES  |     | NULL    |       |
| STATE_PROVINCE | varchar(25)  | YES  |     | NULL    |       |
| COUNTRY_ID     | varchar(2)   | YES  |     | NULL    |       |
| region_id      | int(11)      | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
```

- Write a SQL statement to add a columns ID as the first column of the table locations.

```
ALTER TABLE locations
ADD ID INT FIRST;
```

Sample Output:

```
mysql> SHOW COLUMNS FROM locations;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| ID             | int(11)      | YES  |     | NULL    |       |
| LOCATION_ID    | decimal(4,0) | YES  |     | NULL    |       |
| STREET_ADDRESS | varchar(40)  | YES  |     | NULL    |       |
| POSTAL_CODE    | varchar(12)  | YES  |     | NULL    |       |
| CITY           | varchar(30)  | YES  |     | NULL    |       |
| STATE_PROVINCE | varchar(25)  | YES  |     | NULL    |       |
| COUNTRY_ID     | varchar(2)   | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
```

- Write a SQL statement to add a column region_id after state_province to the table locations.

```
ALTER TABLE locations
ADD region_id INT
AFTER state_province;
```

Sample Output:

```
mysql> SHOW COLUMNS FROM locations;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| LOCATION_ID    | decimal(4,0) | YES  |     | NULL    |       |
| STREET_ADDRESS | varchar(40)  | YES  |     | NULL    |       |
| POSTAL_CODE    | varchar(12)  | YES  |     | NULL    |       |
| CITY           | varchar(30)  | YES  |     | NULL    |       |
| STATE_PROVINCE | varchar(25)  | YES  |     | NULL    |       |
| region_id      | int(11)      | YES  |     | NULL    |       |
| COUNTRY_ID     | varchar(2)   | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

- Write a SQL statement change the data type of the column country_id to integer in the table locations.

```
ALTER TABLE locations
MODIFY country_id INT;
```

Sample Output:

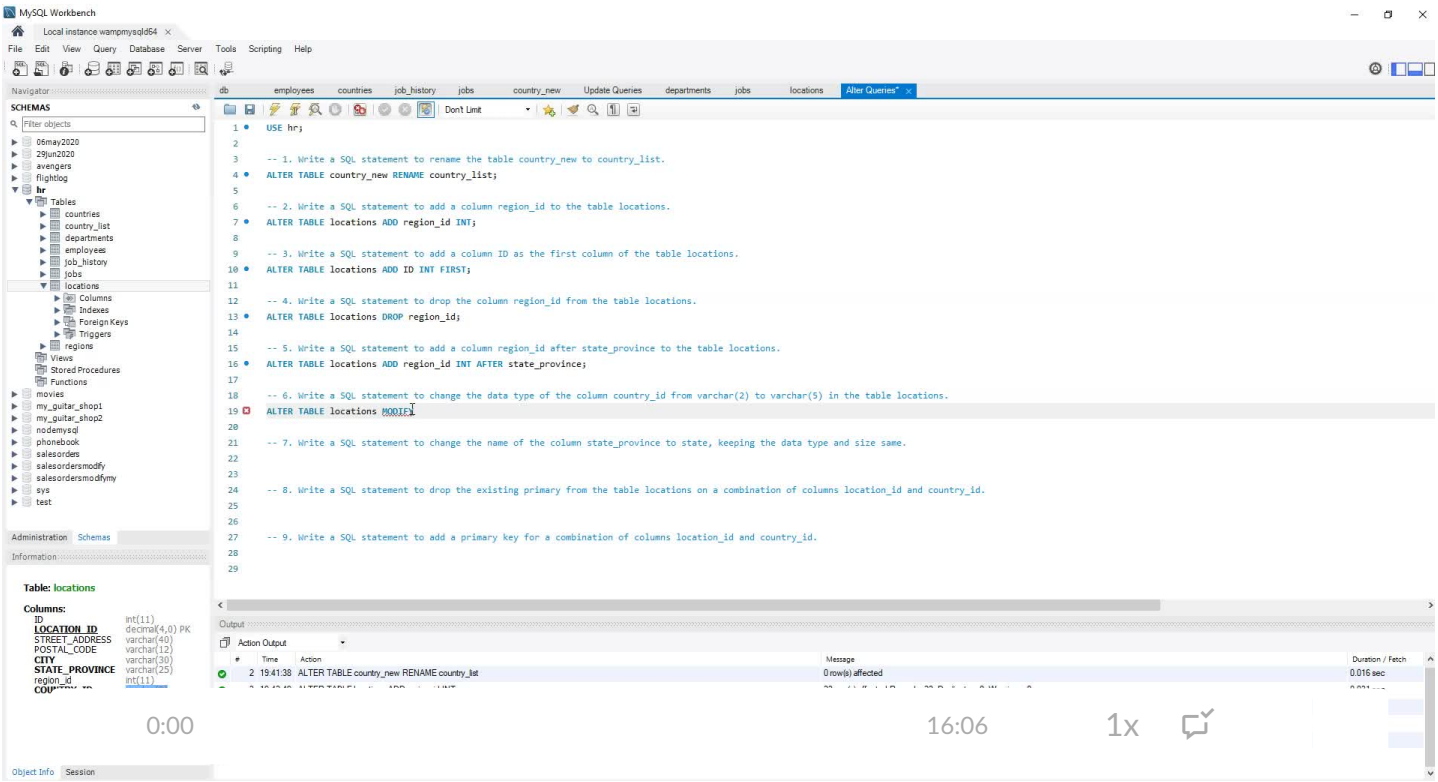
```
mysql> SHOW COLUMNS FROM locations;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| LOCATION_ID    | decimal(4,0) | YES  |     | NULL    |       |
| STREET_ADDRESS | varchar(40)  | YES  |     | NULL    |       |
| POSTAL_CODE    | varchar(12)  | YES  |     | NULL    |       |
| CITY           | varchar(30)  | YES  |     | NULL    |       |
| STATE_PROVINCE | varchar(25)  | YES  |     | NULL    |       |
| country_id     | int(11)      | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

- Write a SQL statement to drop the column city from the table locations.

```
ALTER TABLE locations
DROP city;
```

Sample Output:

```
mysql> SHOW COLUMNS FROM locations;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| LOCATION_ID    | decimal(4,0) | YES  |     | NULL    |       |
| STREET_ADDRESS | varchar(40)  | YES  |     | NULL    |       |
| POSTAL_CODE    | varchar(12)  | YES  |     | NULL    |       |
| STATE_PROVINCE | varchar(25)  | YES  |     | NULL    |       |
| COUNTRY_ID     | varchar(2)   | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```



- Write a SQL statement to rename the table countries to country_new.

```
ALTER TABLE countries RENAME country_new;
```

Sample Output:

```
+---------------+
| Tables_in_hrr |
+---------------+
| country_new   |
| departments   |
| dup_countries |
| employees     |
| jobs          |
+---------------+
```

- Write a SQL statement to add a column region_id to the table locations.

```
ALTER TABLE locations
ADD region_id  INT;
```

Sample Output:

- Write a SQL statement to change the name of the column state_province to state, keeping the data type and size same.

```
ALTER TABLE locations
DROP state_province,
ADD state varchar(25)
AFTER city;
```

In that case, if there are no data in the table, the old column will be removed and new column will be created, no problem at all, but if data in the table exists, you can use the following statement:

```
ALTER TABLE locations
     CHANGE state_province state varchar(25);
```

Sample Output:

```
mysql> SHOW COLUMNS FROM locations;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| LOCATION_ID    | decimal(4,0) | YES  |     | NULL    |       |
| STREET_ADDRESS | varchar(40)  | YES  |     | NULL    |       |
| POSTAL_CODE    | varchar(12)  | YES  |     | NULL    |       |
| CITY           | varchar(30)  | YES  |     | NULL    |       |
| state          | varchar(25)  | YES  |     | NULL    |       |
| COUNTRY_ID     | varchar(2)   | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
```

- Write a SQL statement to add a primary key for the columns location_id in the locations table.

```
ALTER TABLE locations
ADD PRIMARY KEY(location_id);
```

Sample Output:

```
mysql> SHOW COLUMNS FROM locations;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| LOCATION_ID    | decimal(4,0) | NO   | PRI | 0       |       |
| STREET_ADDRESS | varchar(40)  | YES  |     | NULL    |       |
| POSTAL_CODE    | varchar(12)  | YES  |     | NULL    |       |
| CITY           | varchar(30)  | YES  |     | NULL    |       |
| STATE_PROVINCE | varchar(25)  | YES  |     | NULL    |       |
| COUNTRY_ID     | varchar(2)   | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
```

- Write a SQL statement to add a primary key for a combination of columns location_id and country_id.

```
ALTER TABLE locations
ADD PRIMARY KEY(location_id,country_id);
```

Now see the structure of the table locations after alteration.

Sample Output:

```
mysql> SHOW COLUMNS FROM locations;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| LOCATION_ID    | decimal(4,0) | NO   | PRI | 0       |       |
| STREET_ADDRESS | varchar(40)  | YES  |     | NULL    |       |
| POSTAL_CODE    | varchar(12)  | YES  |     | NULL    |       |
| CITY           | varchar(30)  | YES  |     | NULL    |       |
| STATE_PROVINCE | varchar(25)  | YES  |     | NULL    |       |
| COUNTRY_ID     | varchar(2)   | NO   | PRI |         |       |
+----------------+--------------+------+-----+---------+-------+
```

Here is the index file which have been created with creation of primary key.

```
+-----------+------------+----------+--------------+-------------+-----------+-------------+----------+----
| Table     | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Pac
+-----------+------------+----------+--------------+-------------+-----------+-------------+----------+----
| locations |          0 | PRIMARY  |            1 | LOCATION_ID | A         |        NULL |     NULL | NUL
| locations |          0 | PRIMARY  |            2 | COUNTRY_ID  | A         |           0 |     NULL | NUL
+-----------+------------+----------+--------------+-------------+-----------+-------------+----------+----
```

- Write a SQL statement to drop the existing primary from the table locations on a combination of columns location_id and country_id.

```
ALTER TABLE locations DROP PRIMARY KEY;
```

Sample Output:

```
mysql> SHOW COLUMNS FROM locations;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| LOCATION_ID    | decimal(4,0) | NO   |     | NULL    |       |
| STREET_ADDRESS | varchar(40)  | YES  |     | NULL    |       |
| POSTAL_CODE    | varchar(12)  | YES  |     | NULL    |       |
| CITY           | varchar(30)  | YES  |     | NULL    |       |
| STATE_PROVINCE | varchar(25)  | YES  |     | NULL    |       |
| COUNTRY_ID     | varchar(2)   | NO   |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
```

- Write a SQL statement to add a foreign key on job_id column of job_history table referencing to the primary key job_id of jobs table.

```
ALTER TABLE job_history
ADD FOREIGN KEY(job_id)
REFERENCES jobs(job_id);
```

Let execute the above code in MySQL 5.6 command prompt

Now see the structure of the table job_history after being altered.

Sample Output:

```
mysql> SHOW COLUMNS FROM job_history;
+---------------+---------+------+-----+---------+-------+
| Field         | Type    | Null | Key | Default | Extra |
+---------------+---------+------+-----+---------+-------+
| EMPLOYEE_ID   | int(11) | NO   |     | NULL    |       |
| START_DATE    | date    | YES  |     | NULL    |       |
| HIRE_DATE     | date    | YES  |     | NULL    |       |
| JOB_ID        | int(11) | NO   | MUL | NULL    |       |
| DEPARTMENT_ID | int(11) | NO   |     | NULL    |       |
+---------------+---------+------+-----+---------+-------+
```

Now see the created index file.

```
mysql> SHOW INDEX FROM job_history;
+-------------+------------+----------+--------------+-------------+-----------+-------------+----------+---
| Table       | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Pa
+-------------+------------+----------+--------------+-------------+-----------+-------------+----------+---
| job_history |          1 | JOB_ID   |            1 | JOB_ID      | A         |        NULL |     NULL | NU
+-------------+------------+----------+--------------+-------------+-----------+-------------+----------+---
```

- Write a SQL statement to add a foreign key constraint named fk_job_id on job_id column of job_history table referencing to the primary key job_id of jobs table.

```
ALTER TABLE job_history
ADD CONSTRAINT fk_job_id
FOREIGN KEY (job_id)
REFERENCES jobs(job_id)
ON UPDATE RESTRICT
ON DELETE CASCADE;
```

Now see the structure of the table locations after being altered.

```
mysql> SHOW COLUMNS FROM job_history;
+---------------+---------+------+-----+---------+-------+
| Field         | Type    | Null | Key | Default | Extra |
+---------------+---------+------+-----+---------+-------+
| EMPLOYEE_ID   | int(11) | NO   |     | NULL    |       |
| START_DATE    | date    | YES  |     | NULL    |       |
| HIRE_DATE     | date    | YES  |     | NULL    |       |
| JOB_ID        | int(11) | NO   | MUL | NULL    |       |
| DEPARTMENT_ID | int(11) | NO   |     | NULL    |       |
+---------------+---------+------+-----+---------+-------+
```

Now see the created index file.

```
mysql> SHOW INDEX FROM job_history;
+-------------+------------+-----------+--------------+-------------+-----------+-------------+----------+----
| Table       | Non_unique | Key_name  | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Pac
+-------------+------------+-----------+--------------+-------------+-----------+-------------+----------+----
| job_history |          1 | fk_job_id |            1 | JOB_ID      | A         |        NULL |     NULL | NUL
+-------------+------------+-----------+--------------+-------------+-----------+-------------+----------+----
```

- Write a SQL statement to drop the existing foreign key fk_job_id from job_history table on job_id column which is referencing to the job_id of jobs table.

```
ALTER TABLE job_history
DROP FOREIGN KEY fk_job_id;
```

Now see the structure of the table job_history after being altered.

```
mysql> SHOW COLUMNS FROM job_history;
+---------------+---------+------+-----+---------+-------+
| Field         | Type    | Null | Key | Default | Extra |
+---------------+---------+------+-----+---------+-------+
| EMPLOYEE_ID   | int(11) | NO   |     | NULL    |       |
| START_DATE    | date    | YES  |     | NULL    |       |
| HIRE_DATE     | date    | YES  |     | NULL    |       |
| JOB_ID        | int(11) | NO   | MUL | NULL    |       |
| DEPARTMENT_ID | int(11) | NO   |     | NULL    |       |
+---------------+---------+------+-----+---------+-------+
```

Now see the index file.

```
+-------------+------------+-----------+--------------+-------------+-----------+-------------+----------+---
| Table       | Non_unique | Key_name  | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Pa
+-------------+------------+-----------+--------------+-------------+-----------+-------------+----------+---
| JOB_HISTORY |          1 | fk_job_id |            1 | JOB_ID      | A         |           1 |     NULL | NU
+-------------+------------+-----------+--------------+-------------+-----------+-------------+----------+---
```

- Write a SQL statement to add an index named indx_job_id on job_id column in the table job_history.

```
ALTER TABLE job_history
ADD INDEX indx_job_id(job_id);
```

Now see the structure of the table job_history after being altered.

Sample Output:

```
mysql> SHOW COLUMNS FROM job_history;
+---------------+---------+------+-----+---------+-------+
| Field         | Type    | Null | Key | Default | Extra |
+---------------+---------+------+-----+---------+-------+
| EMPLOYEE_ID   | int(11) | NO   | PRI | NULL    |       |
| START_DATE    | date    | YES  |     | NULL    |       |
| HIRE_DATE     | date    | YES  |     | NULL    |       |
| JOB_ID        | int(11) | NO   | MUL | NULL    |       |
| DEPARTMENT_ID | int(11) | NO   |     | NULL    |       |
+---------------+---------+------+-----+---------+-------+
```

Now see the index file.

```
mysql> SHOW INDEXES FROM job_history;
+-------------+------------+-------------+--------------+-------------+-----------+-------------+----------+---
| Table       | Non_unique | Key_name    | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Pa
+-------------+------------+-------------+--------------+-------------+-----------+-------------+----------+---
| job_history |          0 | PRIMARY     |            1 | EMPLOYEE_ID | A         |           0 |     NULL | NU
| job_history |          1 | indx_job_id |            1 | JOB_ID      | A         |           0 |     NULL | NU
+-------------+------------+-------------+--------------+-------------+-----------+-------------+----------+---
```

- Write a SQL statement to drop the index indx_job_id from job_history table.

```
ALTER TABLE job_history
DROP INDEX indx_job_id;
```

Now see the structure of the table job_history after being altered.

```
mysql> SHOW COLUMNS FROM job_history;
+---------------+---------+------+-----+---------+-------+
| Field         | Type    | Null | Key | Default | Extra |
+---------------+---------+------+-----+---------+-------+
| EMPLOYEE_ID   | int(11) | NO   | PRI | NULL    |       |
| START_DATE    | date    | YES  |     | NULL    |       |
| HIRE_DATE     | date    | YES  |     | NULL    |       |
| JOB_ID        | int(11) | NO   |     | NULL    |       |
| DEPARTMENT_ID | int(11) | NO   |     | NULL    |       |
+---------------+---------+------+-----+---------+-------+
```

Now see the index file.

```
mysql> SHOW INDEXES FROM job_history;
+-------------+------------+----------+--------------+-------------+-----------+-------------+----------+----
| Table       | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Pac
+-------------+------------+----------+--------------+-------------+-----------+-------------+----------+----
| job_history |          0 | PRIMARY  |            1 | EMPLOYEE_ID | A         |           0 |     NULL | NUL
+-------------+------------+----------+--------------+-------------+-----------+-------------+----------+----
```