

The image features a solid teal background. A white rectangular area is positioned on the right side, containing text. Above the text is a teal rectangular block. Below the text is a teal horizontal line.

Java Enterprise Edition

ANUDIP FOUNDATION



MySQL insert into Statement: Exercise-1

Time: 180 min

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

Here in the following is the structure of the table countries.

Field	Type	Null	Key	Default	Extra
COUNTRY_ID	varchar(2)	YES		NULL	
COUNTRY_NAME	varchar(40)	YES		NULL	
REGION_ID	decimal(10,0)	YES		NULL	

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

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

MySQL insert into Statement: Exercise-2

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

Here in the following is the structure of the table countries.

Field	Type	Null	Key	Default	Extra
COUNTRY_ID	varchar(2)	YES		NULL	
COUNTRY_NAME	varchar(40)	YES		NULL	
REGION_ID	decimal(10,0)	YES		NULL	

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

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

MySQL insert into Statement: Exercise-3

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

Here in the following is the structure of the table countries.

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| COUNTRY_ID | varchar(2) | YES  |     | NULL    |      |
| COUNTRY_NAME | varchar(40) | YES  |     | NULL    |      |
| REGION_ID   | decimal(10,0) | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

Sample Solution:

```
CREATE TABLE IF NOT EXISTS country_new
```

```
AS SELECT * FROM countries;
```

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

```
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)
```

MySQL insert into Statement: Exercise-4

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

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

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

MySQL insert into Statement: Exercise-5

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

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

```
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)

MySQL insert into Statement: Exercise-6

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

Here is the rows for country_new table. Assume that, the countries table is empty.

```
+-----+-----+-----+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+-----+-----+-----+
| C0001      | India        | 1001      |
| C0002      | USA          | 1007      |
| C0003      | UK           | 1003      |
+-----+-----+-----+
```

Sample Solution:

```
INSERT INTO countries
```

```
SELECT * FROM country_new;
```

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

```
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)
```

MySQL insert into Statement: Exercise-7

7. 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);
```

```
mysql> SELECT * FROM jobs;
```

```

+-----+-----+-----+
| JOB_ID | JOB_TITLE | MIN_SALARY |
+-----+-----+-----+
| 1001 | OFFICER | 8000 |
+-----+-----+-----+

```

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

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

MySQL insert into Statement: Exercise-8

8. 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 the table jobs.

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

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

```
mysql> SELECT * FROM jobs;
```

```

+-----+-----+-----+
| JOB_ID | JOB_TITLE | MIN_SALARY |
+-----+-----+-----+
| 1001 | OFFICER | 8000 |
+-----+-----+-----+

```

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

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

MySQL insert into Statement: Exercise-9

9. 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.

Create the table countries.

```
CREATE TABLE IF NOT EXISTS countries (
  COUNTRY_ID integer NOT NULL,
  COUNTRY_NAME varchar(40) NOT NULL,
  REGION_ID integer NOT NULL,
  PRIMARY KEY (COUNTRY_ID,REGION_ID)
);
```

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

```
mysql> SELECT * FROM countries;
+-----+-----+-----+
| COUNTRY_ID | COUNTRY_NAME | REGION_ID |
+-----+-----+-----+
|      501 | India      |      185 |
+-----+-----+-----+
```

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

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

MySQL insert into Statement: Exercise-10

10. 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.

```
CREATE TABLE IF NOT EXISTS countries (
  COUNTRY_ID integer NOT NULL AUTO_INCREMENT PRIMARY KEY,
  COUNTRY_NAME varchar(40) NOT NULL,
  REGION_ID integer NOT NULL
);
```

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

```
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);
```

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

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

MySQL insert into Statement: Exercise-11

11. 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.

Create the table countries.

```
CREATE TABLE IF NOT EXISTS countries (
COUNTRY_ID integer NOT NULL AUTO_INCREMENT PRIMARY KEY,
COUNTRY_NAME varchar(40) NOT NULL DEFAULT 'N/A',
REGION_ID integer NOT NULL
);
```

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

```
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);
```

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

```
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);
```

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

```
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)
```

MySQL insert into Statement: Exercise-12

12. 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.

Sample table jobs.

```
CREATE TABLE IF NOT EXISTS jobs (
  JOB_ID integer NOT NULL UNIQUE PRIMARY KEY,
  JOB_TITLE varchar(35) NOT NULL DEFAULT ' ',
  MIN_SALARY decimal(6,0) DEFAULT 8000,
  MAX_SALARY decimal(6,0) DEFAULT 20000
)ENGINE=InnoDB;
```

```
INSERT INTO jobs(JOB_ID,JOB_TITLE) VALUES(1001,'OFFICER');
INSERT INTO jobs(JOB_ID,JOB_TITLE) VALUES(1002,'CLERK');
```

```
+-----+-----+-----+-----+
|
```

```
| JOB_ID | JOB_TITLE | MIN_SALARY | MAX_SALARY |
+-----+-----+-----+-----+
| 1001 | OFFICER | 8000 | 20000 |
| 1002 | CLERK | 8000 | 20000 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Sample table job_history;

```
CREATE TABLE job_history (
EMPLOYEE_ID integer NOT NULL PRIMARY KEY,
JOB_ID integer NOT NULL,
DEPARTMENT_ID integer DEFAULT NULL,
FOREIGN KEY (job_id) REFERENCES jobs(job_id)
)ENGINE=InnoDB;
```

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

```
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 arise.

Now insert another row in the job_history table.

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

```
mysql> INSERT INTO job_history VALUES(502,1003,80);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`hr`.`job_history`, CONSTRAINT `job_history_ibfk_1`
(`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 can not contain the value of job_id as specified. Here the primary key - foreign key relationship is violating and shows the above message.

MySQL insert into Statement: Exercise-13

13. 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.

Sample table departments.

```
CREATE TABLE IF NOT EXISTS departments (
DEPARTMENT_ID integer NOT NULL UNIQUE,
DEPARTMENT_NAME varchar(30) NOT NULL,
MANAGER_ID integer DEFAULT NULL,
LOCATION_ID integer DEFAULT NULL,
PRIMARY KEY (DEPARTMENT_ID,MANAGER_ID)
)ENGINE=InnoDB;
```

```
INSERT INTO departments VALUES(60,'SALES',201,89);
INSERT INTO departments VALUES(61,'ACCOUNTS',201,89);
INSERT INTO departments VALUES(80,'FINANCE',211,90);
```

```
mysql> SELECT * FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
60	SALES	201	89
61	ACCOUNTS	201	89
80	FINANCE	211	90

```
3 rows in set (0.00 sec)
```

Sample table employees.

```
CREATE TABLE IF NOT EXISTS employees (
EMPLOYEE_ID integer NOT NULL PRIMARY KEY,
FIRST_NAME varchar(20) DEFAULT NULL,
LAST_NAME varchar(25) NOT NULL,
JOB_ID varchar(10) NOT NULL,
SALARY decimal(8,2) DEFAULT NULL,
MANAGER_ID integer DEFAULT NULL,
DEPARTMENT_ID integer DEFAULT NULL,
FOREIGN KEY(DEPARTMENT_ID,MANAGER_ID)
REFERENCES departments(DEPARTMENT_ID,MANAGER_ID)
)ENGINE=InnoDB;
```

Now insert the rows in the employees.

Sample Solution:

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

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

```
mysql> SELECT * FROM employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID
510	Alex	Hanes	CLERK	18000.00	201	60
511	Kim	Leon	CLERK	18000.00	211	80

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);
```

Copy

Let execute the above code in MySQL 5.6 command prompt.

```
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
(`hr`.`employees`, CONSTRAINT `employees_ibfk_1` FOREIGN KEY (`DEPARTMENT_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 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.

MySQL insert into Statement: Exercise-14

14. 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.

Sample table departments.

```
CREATE TABLE IF NOT EXISTS departments (
  DEPARTMENT_ID integer NOT NULL UNIQUE,
  DEPARTMENT_NAME varchar(30) NOT NULL,
  MANAGER_ID integer DEFAULT NULL,
  LOCATION_ID integer DEFAULT NULL,
  PRIMARY KEY (DEPARTMENT_ID)
)ENGINE=InnoDB;
```

```
INSERT INTO departments VALUES(60,'SALES',201,89);
INSERT INTO departments VALUES(61,'ACCOUNTS',201,89);
```

```
mysql> SELECT * FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
60	SALES	201	89
61	ACCOUNTS	201	89

```
2 rows in set (0.00 sec)
```

Sample table jobs.

```
CREATE TABLE IF NOT EXISTS jobs (
JOB_ID integer NOT NULL UNIQUE PRIMARY KEY,
JOB_TITLE varchar(35) NOT NULL DEFAULT ' ',
MIN_SALARY decimal(6,0) DEFAULT 8000,
MAX_SALARY decimal(6,0) DEFAULT 20000
)ENGINE=InnoDB;
```

```
INSERT INTO jobs(JOB_ID,JOB_TITLE) VALUES(1001,'OFFICER');
INSERT INTO jobs(JOB_ID,JOB_TITLE) VALUES(1002,'CLERK');
```

```
mysql> SELECT * FROM jobs;
```

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
1001	OFFICER	8000	20000
1002	CLERK	8000	20000

```
2 rows in set (0.00 sec)
```

Sample table employees.

```
CREATE TABLE IF NOT EXISTS employees (
EMPLOYEE_ID integer NOT NULL PRIMARY KEY,
FIRST_NAME varchar(20) DEFAULT NULL,
LAST_NAME varchar(25) NOT NULL,
DEPARTMENT_ID integer DEFAULT NULL,
FOREIGN KEY(DEPARTMENT_ID)
REFERENCES departments(DEPARTMENT_ID),
JOB_ID integer NOT NULL,
FOREIGN KEY(JOB_ID)
REFERENCES jobs(JOB_ID),
SALARY decimal(8,2) DEFAULT NULL
)ENGINE=InnoDB;
```

Now insert the rows into the table employees.

Sample Solution:

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

Copy

Let execute the above code in MySQL 5.6 command prompt.

Here is the structure of the table:

```
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);
```

Copy

Let execute the above code in MySQL 5.6 command prompt.

```
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`hr`.`employees`, CONSTRAINT `employees_ibfk_2` FOREIGN
KEY (`JOB_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);
```

Copy

```
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`hr`.`employees`, CONSTRAINT `employees_ibfk_2` FOREIGN
KEY (`JOB_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.

