

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 Alter Table

Time: 90 Min

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

Here is the list of tables.

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

Code:

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

Let execute the above code in MySQL 5.6 command prompt

Now, after execute the command see the list of tables.

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

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

Here is the structure of the table locations.

```
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    |       |
```

+-----+-----+-----+-----+-----+
Code:

```
ALTER TABLE locations
```

```
ADD region_id INT;
```

Let execute the above code in MySQL 5.6 command prompt

See the structure of the table after alteration.

```
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	

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

Here is the structure of the table locations.

```
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	

Code:

```
ALTER TABLE locations
```

```
ADD ID INT FIRST;
```

Let execute the above code in MySQL 5.6 command prompt

See the structure of the table after alteration.

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

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

Here is the structure of the table locations.

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

Code:

```
ALTER TABLE locations
```

```
ADD region_id INT
```

```
AFTER state_province;
```

Let execute the above code in MySQL 5.6 command prompt

Here is structure of the table locations after alteration.

```
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 | |
| COUNTRY_ID | varchar(2) | YES | | NULL | |
+-----+-----+-----+-----+

```

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

Here is the structure of the table locations.

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

```

Code:

```
ALTER TABLE locations
```

```
ADD region_id INT
```

```
AFTER state_province;
```

Let execute the above code in MySQL 5.6 command prompt

Here is structure of the table locations after alteration.

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

```

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

Here is the structure of the table locations.

```
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	

Code:

```
ALTER TABLE locations
```

```
DROP city;
```

Let execute the above code in MySQL 5.6 command prompt

Now see the structure of the table locations after alteration.

```
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	

The above result shows that there is no 'city' column; it has been dropped from the table locations.

7. Write a SQL statement to change the name of the column state_province to state, keeping the data type Here is the structure of the table locations.

```
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	

Code:

```
ALTER TABLE locations
```

```
DROP state_province,
```

```
ADD state varchar(25)
```

```
AFTER city;
```

Let execute the above code in MySQL 5.6 command prompt

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

```
ALTER TABLE locations
```

```
CHANGE state_province state varchar(25);
```

Now see the structure of the table locations after alteration.

Here is the structure of the table locations.

```
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	

Code:

```
ALTER TABLE locations
```

```
DROP state_province,
```

```
ADD state varchar(25)
```

```
AFTER city;
```

Let execute the above code in MySQL 5.6 command prompt

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

```
ALTER TABLE locations
```

CHANGE state_province state **varchar(25)**;

Now see the structure of the table locations after alteration.

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

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

Here is the sample table employees.

Sample table: employees

```
+-----+-----+-----+-----+-----+-----+-----+-----+
--+-----+-----+-----+-----+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | HIRE_DATE |
JOB_ID | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
+-----+-----+-----+-----+-----+-----+-----+-----+
--+-----+-----+-----+-----+
| 100 | Steven | King | SKING | 515.123.4567 | 1987-06-17 | AD_PRES |
24000.00 | 0.00 | 0 | 90 |
| 101 | Neena | Kochhar | NKOCHHAR | 515.123.4568 | 1987-06-18 | AD_VP |
17000.00 | 0.00 | 100 | 90 |
| 102 | Lex | De Haan | LDEHAAN | 515.123.4569 | 1987-06-19 | AD_VP |
17000.00 | 0.00 | 100 | 90 |
| 103 | Alexander | Hunold | AHUNOLD | 590.423.4567 | 1987-06-20 | IT_PROG |
9000.00 | 0.00 | 102 | 60 |
| 104 | Bruce | Ernst | BERNST | 590.423.4568 | 1987-06-21 | IT_PROG |
6000.00 | 0.00 | 103 | 60 |
| 105 | David | Austin | DAUSTIN | 590.423.4569 | 1987-06-22 | IT_PROG |
4800.00 | 0.00 | 103 | 60 |
| 106 | Valli | Pataballa | VPATABAL | 590.423.4560 | 1987-06-23 | IT_PROG |
4800.00 | 0.00 | 103 | 60 |
| 107 | Diana | Lorentz | DLORENTZ | 590.423.5567 | 1987-06-24 | IT_PROG |
4200.00 | 0.00 | 103 | 60 |
| 108 | Nancy | Greenberg | NGREENBE | 515.124.4569 | 1987-06-25 | FI_MGR |
12000.00 | 0.00 | 101 | 100 |
| 109 | Daniel | Faviat | DFAVIET | 515.124.4169 | 1987-06-26 | FI_ACCOUNT |
9000.00 | 0.00 | 108 | 100 |
| 110 | John | Chen | JCHEN | 515.124.4269 | 1987-06-27 | FI_ACCOUNT |
8200.00 | 0.00 | 108 | 100 |
```


	111	Ismael	Sciarra	ISCIARRA	515.124.4369	1987-06-28	FI_ACCOUNT	
7700.00		0.00	108	100				
	112	Jose Manuel	Urman	JMURMAN	515.124.4469	1987-06-29	FI_ACCOUNT	
7800.00		0.00	108	100				
	113	Luis	Popp	LPOPP	515.124.4567	1987-06-30	FI_ACCOUNT	
6900.00		0.00	108	100				
	114	Den	Raphaely	DRAPHEAL	515.127.4561	1987-07-01	PU_MAN	
11000.00		0.00	100	30				
	115	Alexander	Khoo	AKHOO	515.127.4562	1987-07-02	PU_CLERK	
3100.00		0.00	114	30				
	116	Shelli	Baida	SBAIDA	515.127.4563	1987-07-03	PU_CLERK	
2900.00		0.00	114	30				
	117	Sigal	Tobias	STOBIAS	515.127.4564	1987-07-04	PU_CLERK	
2800.00		0.00	114	30				
	118	Guy	Himuro	GHIMURO	515.127.4565	1987-07-05	PU_CLERK	
2600.00		0.00	114	30				
	119	Karen	Colmenares	KCOLMENA	515.127.4566	1987-07-06	PU_CLERK	
2500.00		0.00	114	30				
	120	Matthew	Weiss	MWEISS	650.123.1234	1987-07-07	ST_MAN	
8000.00		0.00	100	50				
	121	Adam	Fripp	AFRIPP	650.123.2234	1987-07-08	ST_MAN	
8200.00		0.00	100	50				
	122	Payam	Kaufling	PKAUFLIN	650.123.3234	1987-07-09	ST_MAN	
7900.00		0.00	100	50				
	123	Shanta	Vollman	SVOLLMAN	650.123.4234	1987-07-10	ST_MAN	
6500.00		0.00	100	50				
	124	Kevin	Mourgos	KMOURGOS	650.123.5234	1987-07-11	ST_MAN	
5800.00		0.00	100	50				
	125	Julia	Nayer	JNAYER	650.124.1214	1987-07-12	ST_CLERK	
3200.00		0.00	120	50				
	126	Irene	Mikkilineni	IMIKKILI	650.124.1224	1987-07-13	ST_CLERK	
2700.00		0.00	120	50				
	127	James	Landry	JLANDRY	650.124.1334	1987-07-14	ST_CLERK	
2400.00		0.00	120	50				
	128	Steven	Markle	SMARKLE	650.124.1434	1987-07-15	ST_CLERK	
2200.00		0.00	120	50				
	129	Laura	Bissot	LBISSOT	650.124.5234	1987-07-16	ST_CLERK	
3300.00		0.00	121	50				
	130	Mozhe	Atkinson	MATKINSO	650.124.6234	1987-07-17	ST_CLERK	
2800.00		0.00	121	50				
	131	James	Marlow	JAMRLOW	650.124.7234	1987-07-18	ST_CLERK	
2500.00		0.00	121	50				
	132	TJ	Olson	TJOLSON	650.124.8234	1987-07-19	ST_CLERK	
2100.00		0.00	121	50				
	133	Jason	Mallin	JMALLIN	650.127.1934	1987-07-20	ST_CLERK	
3300.00		0.00	122	50				
	134	Michael	Rogers	MROGERS	650.127.1834	1987-07-21	ST_CLERK	
2900.00		0.00	122	50				
	135	Ki	Gee	KGEE	650.127.1734	1987-07-22	ST_CLERK	2400.00
	0.00		122	50				

136	Hazel	Philtanker	HPHILTAN	650.127.1634	1987-07-23	ST_CLERK
2200.00	0.00	122	50			
137	Renske	Ladwig	RLADWIG	650.121.1234	1987-07-24	ST_CLERK
3600.00	0.00	123	50			
138	Stephen	Stiles	SSTILES	650.121.2034	1987-07-25	ST_CLERK
3200.00	0.00	123	50			
139	John	Seo	JSEO	650.121.2019	1987-07-26	ST_CLERK
2700.00	0.00	123	50			
140	Joshua	Patel	JPATEL	650.121.1834	1987-07-27	ST_CLERK
2500.00	0.00	123	50			
141	Trenna	Rajs	TRAJS	650.121.8009	1987-07-28	ST_CLERK
3500.00	0.00	124	50			
142	Curtis	Davies	CDAVIES	650.121.2994	1987-07-29	ST_CLERK
3100.00	0.00	124	50			
143	Randall	Matos	RMATOS	650.121.2874	1987-07-30	ST_CLERK
2600.00	0.00	124	50			
144	Peter	Vargas	PVARGAS	650.121.2004	1987-07-31	ST_CLERK
2500.00	0.00	124	50			
145	John	Russell	JRUSSEL	011.44.1344.429268	1987-08-01	SA_MAN
14000.00	0.40	100	80			
146	Karen	Partners	KPARTNER	011.44.1344.467268	1987-08-02	SA_MAN
13500.00	0.30	100	80			
147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	1987-08-03	SA_MAN
12000.00	0.30	100	80			
148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	1987-08-04	SA_MAN
11000.00	0.30	100	80			
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	1987-08-05	SA_MAN
10500.00	0.20	100	80			
150	Peter	Tucker	PTUCKER	011.44.1344.129268	1987-08-06	SA_REP
10000.00	0.30	145	80			
151	David	Bernstein	DBERNSTE	011.44.1344.345268	1987-08-07	SA_REP
9500.00	0.25	145	80			
152	Peter	Hall	PHALL	011.44.1344.478968	1987-08-08	SA_REP
9000.00	0.25	145	80			
153	Christopher	Olsen	COLSEN	011.44.1344.498718	1987-08-09	SA_REP
8000.00	0.20	145	80			
154	Nanette	Cambrault	NCAMBRAU	011.44.1344.987668	1987-08-10	SA_REP
7500.00	0.20	145	80			
155	Oliver	Tuvault	OTUVAULT	011.44.1344.486508	1987-08-11	SA_REP
7000.00	0.15	145	80			
156	Janette	King	JKING	011.44.1345.429268	1987-08-12	SA_REP
10000.00	0.35	146	80			
157	Patrick	Sully	PSULLY	011.44.1345.929268	1987-08-13	SA_REP
9500.00	0.35	146	80			
158	Allan	McEwen	AMCEWEN	011.44.1345.829268	1987-08-14	SA_REP
9000.00	0.35	146	80			
159	Lindsey	Smith	LSMITH	011.44.1345.729268	1987-08-15	SA_REP
8000.00	0.30	146	80			
160	Louise	Doran	LDORAN	011.44.1345.629268	1987-08-16	SA_REP
7500.00	0.30	146	80			

161	Sarath	Sewall	SSEWALL	011.44.1345.529268	1987-08-17	SA_REP
7000.00	0.25	146	80			
162	Clara	Vishney	CVISHNEY	011.44.1346.129268	1987-08-18	SA_REP
10500.00	0.25	147	80			
163	Danielle	Greene	DGREENE	011.44.1346.229268	1987-08-19	SA_REP
9500.00	0.15	147	80			
164	Mattea	Marvins	MMARVINS	011.44.1346.329268	1987-08-20	SA_REP
7200.00	0.10	147	80			
165	David	Lee	DLEE	011.44.1346.529268	1987-08-21	SA_REP
6800.00	0.10	147	80			
166	Sundar	Ande	SANDE	011.44.1346.629268	1987-08-22	SA_REP
6400.00	0.10	147	80			
167	Amit	Banda	ABANDA	011.44.1346.729268	1987-08-23	SA_REP
6200.00	0.10	147	80			
168	Lisa	Ozer	LOZER	011.44.1343.929268	1987-08-24	SA_REP
11500.00	0.25	148	80			
169	Harrison	Bloom	HBLOOM	011.44.1343.829268	1987-08-25	SA_REP
10000.00	0.20	148	80			
170	Tayler	Fox	TFOX	011.44.1343.729268	1987-08-26	SA_REP
9600.00	0.20	148	80			
171	William	Smith	WSMITH	011.44.1343.629268	1987-08-27	SA_REP
7400.00	0.15	148	80			
172	Elizabeth	Bates	EBATES	011.44.1343.529268	1987-08-28	SA_REP
7300.00	0.15	148	80			
173	Sundita	Kumar	SKUMAR	011.44.1343.329268	1987-08-29	SA_REP
6100.00	0.10	148	80			
174	Ellen	Abel	EABEL	011.44.1644.429267	1987-08-30	SA_REP
11000.00	0.30	149	80			
175	Alyssa	Hutton	AHUTTON	011.44.1644.429266	1987-08-31	SA_REP
8800.00	0.25	149	80			
176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	1987-09-01	SA_REP
8600.00	0.20	149	80			
177	Jack	Livingston	JLIVINGS	011.44.1644.429264	1987-09-02	SA_REP
8400.00	0.20	149	80			
178	Kimberely	Grant	KGRANT	011.44.1644.429263	1987-09-03	SA_REP
7000.00	0.15	149	0			
179	Charles	Johnson	CJOHNSON	011.44.1644.429262	1987-09-04	SA_REP
6200.00	0.10	149	80			
180	Winston	Taylor	WTAYLOR	650.507.9876	1987-09-05	SH_CLERK
3200.00	0.00	120	50			
181	Jean	Fleur	JFLEUR	650.507.9877	1987-09-06	SH_CLERK
3100.00	0.00	120	50			
182	Martha	Sullivan	MSULLIVA	650.507.9878	1987-09-07	SH_CLERK
2500.00	0.00	120	50			
183	Girard	Geoni	GGEONI	650.507.9879	1987-09-08	SH_CLERK
2800.00	0.00	120	50			
184	Nandita	Sarchand	NSARCHAN	650.509.1876	1987-09-09	SH_CLERK
4200.00	0.00	121	50			
185	Alexis	Bull	ABULL	650.509.2876	1987-09-10	SH_CLERK
4100.00	0.00	121	50			

11

```

+-----+-----+-----+-----+-----+
| 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 | |
+-----+-----+-----+-----+-----+

```

Code:

```
ALTER TABLE locations
```

```
ADD PRIMARY KEY(location_id);
```

Let execute the above code in MySQL 5.6 command prompt

Now see the structure of the table locations after alteration.

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

```

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

Here is the structure of the table locations.

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

```

Code:

```
ALTER TABLE locations
```

```
ADD PRIMARY KEY(location_id,country_id);
```

Let execute the above code in MySQL 5.6 command prompt

Now see the structure of the table locations after alteration.

```
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 | Packed     | Null    | Index_type   | Comment     |
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+
| locations | 0 | PRIMARY | 1 | LOCATION_ID | A        | NULL | NULL | NULL |
| BTREE    |   |         |   |              |          |      |      |      |
| locations | 0 | PRIMARY | 2 | COUNTRY_ID  | A        | 0    | NULL | NULL |
| BTREE    |   |         |   |              |          |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+
```

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

Here is the structure of the table locations.

```
mysql> show columns from locations;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| LOCATION_ID | decimal(4,0) | NO  | PRI | 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  | PRI | NULL     |      |
+-----+-----+-----+-----+-----+-----+
```

Code:

`ALTER TABLE locations DROP PRIMARY KEY;`

Let execute the above code in MySQL 5.6 command prompt

Now see the structure of the table locations after alteration.

```
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	

11. 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.

Here is the structure of the table jobs and job_history.

```
mysql> SHOW COLUMNS FORM jobs;
```

Field	Type	Null	Key	Default	Extra
JOB_ID	int(11)	NO	PRI	NULL	
JOB_TITLE	varchar(35)	NO		NULL	
MIN_SALARY	decimal(6,0)	YES		NULL	
MAX_SALARY	decimal(6,0)	YES		NULL	

```
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		NULL	
DEPARTMENT_ID	int(11)	NO		NULL	

Code:

```
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.

```
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  | Packed    | Null    | Index_type   | Comment     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| job_history |          1 | JOB_ID   |          1 | JOB_ID      | A         | NULL       | NULL | NULL |
| BTREE     |          |         |             |             |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

12. 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.

Here is the structure of the table jobs and job_history.

```
mysql> SHOW COLUMNS FROM jobs;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| JOB_ID     | int(11)    | NO   | PRI | NULL    |      |
| JOB_TITLE  | varchar(35) | NO   |     | NULL    |      |
| MIN_SALARY | decimal(6,0) | YES  |     | NULL    |      |
| MAX_SALARY | decimal(6,0) | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

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



```
| DEPARTMENT_ID | int(11) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
```

Code:

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

Let execute the above code in MySQL 5.6 command prompt

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 | Packed | Null | Index_type | Comment |
+-----+-----+-----+-----+-----+-----+-----+
| job_history | 1 | fk_job_id | 1 | JOB_ID | A | NULL | NULL | NULL |
| BTREE | | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+
```

13. 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.

Here is the structure of the table `job_history`.

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

Code:

```
ALTER TABLE job_history
```

```
DROP FOREIGN KEY fk_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.

```
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 | Packed | Null | Index_type | Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+
| JOB_HISTORY |          1 | fk_job_id |          1 | JOB_ID      | A          |          1 | NULL | NULL |
| BTREE      |          |          |          |             |           |           |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

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

Here is the structure of the table job_history.

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

```

Here is the index file of job_history table.

```
mysql> SHOW INDEXES FROM job_history;
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| Table      | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality |
Sub_part | Packed | Null | Index_type | Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| job_history |          0 | PRIMARY |          1 | EMPLOYEE_ID | A          |          0 | NULL | NULL |
| BTREE      |          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

Code:

```
ALTER TABLE job_history
```

```
ADD INDEX indx_job_id(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.

```
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 | Packed | Null | Index_type | Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

```

| job_history |      0 | PRIMARY |      1 | EMPLOYEE_ID | A |      0 | NULL | NULL
| | BTREE | | | | | | | |
| job_history |      1 | indx_job_id |      1 | JOB_ID | A |      0 | NULL | NULL |
| BTREE | | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+

```

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

Here is the structure of the job_history and index file of the table job_history.

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

```

```
mysql> SHOW INDEXES FROM job_history;
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| Table      | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality |
Sub_part | Packed | Null | Index_type | Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| job_history |      0 | PRIMARY |      1 | EMPLOYEE_ID | A |      0 | NULL | NULL
| | BTREE | | | | | | | |
| job_history |      1 | indx_job_id |      1 | JOB_ID | A |      0 | NULL | NULL |
| BTREE | | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+

```

Code:

```
ALTER TABLE job_history
```

```
DROP INDEX indx_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.

```
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 | Packed | Null | Index_type | Comment |
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+
| job_history | 0 | PRIMARY | 1 | EMPLOYEE_ID | A | 0 | NULL | NULL |
| BTREE | |
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+

```