

# 20CS2016L Database Systems Lab URK21CS11

---

## Ex. No. 1(a) – Creating and Managing Tables

### Aim

To execute DDL commands and get the desired output.

### Description

DDL refers to "Data Definition Language", a subset of SQL statements that change the structure of the database schema in some way, typically by creating, deleting, or modifying schema objects such as databases, tables, and views. Most Impala DDL statements start with the keywords CREATE , DROP , or ALTER .

In this schema, we have four main tables: User, Event, Venue, and Ticket.

#### User table:

Column	Data Type
UserID	NUMBER(10)
Name	VARCHAR2(255)
Email	VARCHAR2(255)
Password	VARCHAR2(255)
Phone	VARCHAR2(20)

#### Event table:

Column	Data Type
EventID	NUMBER(10)
Name	VARCHAR2(255)
Date	DATE
Time	TIMESTAMP
VenueID	NUMBER(10) (Foreign key)
Description	VARCHAR2(500)

#### Venue table:

Column	Data Type
VenueID	NUMBER(10)
Name	VARCHAR2(255)
Address	VARCHAR2(255)
City	VARCHAR2(255)
State	VARCHAR2(255)
Country	VARCHAR2(255)

# 20CS2016L Database Systems Lab URK21CS11

**Ticket table:**

Column	Data Type
TicketID	NUMBER(10)
EventID	NUMBER(10) (Foreign key)
UserID	NUMBER(10)
SeatNumber	VARCHAR2(20)
Price	NUMBER(10, 2)
Status	VARCHAR2(50)

## Queries (DDL Commands)

1. Create User, Event, Venue, and ticket tables based on the given schema.

User table: Create table user\_1128(User\_ID number(10), Name varchar(255), Email varchar(255), Password varchar(255), Phone number(20));

```
SQL> Create table user_1128(User_ID number(10), Name varchar(255), Email varchar(255), Password varchar(255), Phone number(20));
```

Table created.

Event table: Create table event\_1128(Event\_ID number(10), Name varchar(255), Date\_event DATE, Time\_event TIMESTAMP, Venue\_ID number(10), Description varchar(255));

```
SQL> Create table event_1128(Event_ID number(10), Name varchar(255), Date_event DATE, Time_event TIMESTAMP, Venue_ID number(10), Description varchar(255));
```

Table created.

Venue table: Create table venue\_1128(Venue\_ID number(10), Name varchar(255), Address varchar(255), City varchar(255), State varchar(255), Country varchar(255));

```
SQL> Create table venue_1128(Venue_ID number(10), Name varchar(255), Address varchar(255), City varchar(255), State varchar(255), Country varchar(255));
```

Table created.

## 20CS2016L Database Systems Lab URK21CS11

Ticket table: Create table Ticket\_1128(Ticket\_ID number(10), Event\_ID number(10),User\_ID number(10),Seat\_Number number(10),Price number(10,2),Status varchar(255));

```
SQL> Create table Ticket_1128(Ticket_ID number(10), Event_ID number(10),User_ID number(10),Seat_Number number(10),Price number(10,2),Status varchar(255));
```

Table created.

2. Describe the tables.

User Table: desc user\_1128;

```
SQL> desc user_1128
```

Name	Null?	Type
USER_ID		NUMBER(10)
NAME		VARCHAR2(255)
EMAIL		VARCHAR2(255)
PASSWORD		VARCHAR2(255)
PHONE		NUMBER(20)

Event Table: desc event\_1128;

```
SQL> desc event_1128;
```

Name	Null?	Type
EVENT_ID		NUMBER(10)
NAME		VARCHAR2(255)
DATE_EVENT		DATE
TIME_EVENT		TIMESTAMP(6)
VENUE_ID		NUMBER(10)
DESCRIPTION		VARCHAR2(255)

Venue Table: desc venue\_1128;

```
SQL> desc venue_1128;
```

Name	Null?	Type
VENUE_ID		NUMBER(10)
NAME		VARCHAR2(255)
ADDRESS		VARCHAR2(255)
CITY		VARCHAR2(255)
STATE		VARCHAR2(255)
COUNTRY		VARCHAR2(255)

## 20CS2016L Database Systems Lab URK21CS11

---

Ticket Table: desc Ticket\_1128;

```
SQL> desc Ticket_1128;
Name                               Null?      Type
-----
TICKET_ID                          NUMBER(10)
EVENT_ID                           NUMBER(10)
USER_ID                             NUMBER(10)
SEAT_NUMBER                         NUMBER(10)
PRICE                              NUMBER(10,2)
STATUS                             VARCHAR2(255)
```

3. Alter the User table to add a new column Age.

Query: alter table user\_1128 add Age number(10);

```
SQL> alter table user_1128 add Age number(10);
Table altered.
```

4. Drop the newly added column Age.

Query: alter table user\_1128 drop column Age;

```
SQL> alter table user_1128 drop column Age;
Table altered.
```

5. Rename the Venue table to Location

Query: alter table venue\_1128 rename to Location;

```
SQL> alter table venue_1128 rename to Location;
Table altered.
```

## 20CS2016L Database Systems Lab URK21CS11

```
SQL> desc Location;
```

Name	Null?	Type
VENUE_ID		NUMBER(10)
NAME		VARCHAR2(255)
ADDRESS		VARCHAR2(255)
CITY		VARCHAR2(255)
STATE		VARCHAR2(255)
COUNTRY		VARCHAR2(255)

6. Modify the size of the Event table & Description column to 1000.

Query: alter table event\_1128 modify Description varchar(1000);

```
SQL> alter table event_1128 modify Description varchar(1000);
```

```
Table altered.
```

```
SQL> desc event_1128
```

Name	Null?	Type
EVENT_ID		NUMBER(10)
NAME		VARCHAR2(255)
DATE_EVENT		DATE
TIME_EVENT		TIMESTAMP(6)
VENUE_ID		NUMBER(10)
DESCRIPTION		VARCHAR2(1000)

7. Drop the SeatNumber column from the Ticket table

Query: alter table Ticket\_1128 drop column Seat\_Number;

```
SQL> alter table Ticket_1128 drop column Seat_Number;
```

```
Table altered.
```

## 20CS2016L Database Systems Lab URK21CS11

```
SQL> desc Ticket_1128
```

Name	Null?	Type
TICKET_ID		NUMBER(10)
EVENT_ID		NUMBER(10)
USER_ID		NUMBER(10)
PRICE		NUMBER(10,2)
STATUS		VARCHAR2(255)

8. Add a unique constraint on the Email column in the User table

Query: alter table user\_1128 add constraint Email UNIQUE(Email);

```
SQL> alter table user_1128 add constraint Email UNIQUE(Email);
```

Table altered.

```
SQL> desc user_1128
```

Name	Null?	Type
USER_ID		NUMBER(10)
NAME		VARCHAR2(255)
EMAIL		VARCHAR2(255)
PASSWORD		VARCHAR2(255)
PHONE		NUMBER(20)

9. Rename the UserID column in the User table to ID

Query: alter table user\_1128 rename column User\_ID to ID;

```
SQL> alter table user_1128 rename column User_ID to ID;
```

Table altered.

10. Modify the Ticket table to add a column named Barcode with a data type of VARCHAR2(50).

Query: alter table Ticket\_1128 add Barcode varchar2(50);

```
SQL> alter table Ticket_1128 add Barcode varchar2(50);
```

Table altered.

## 20CS2016L Database Systems Lab URK21CS11

---

11. Modify the Name column in the Venue table to increase its maximum length to VARCHAR2(300)

Query: alter table Location modify Name varchar (300);

```
SQL> alter table Location modify Name varchar (300);  
Table altered.
```

12. Add a foreign key constraint on the VenueID column in the Event table, referencing the Venue table.

Query: alter table event\_1128 modify Venue\_ID number references Location (Venue\_ID);

13. Add a CHECK constraint to check whether the UserID is between 101 and 105

Query: alter table user\_1128 add constraint ck check(ID between 101 and 105);

```
SQL> alter table user_1128 add constraint ck check(ID between 101 and 105);  
Table altered.
```

14. Add a unique constraint to Phone column of the User table.

Query: alter table user\_1125 add constraint Phone UNIQUE(Phone);

```
SQL> alter table user_1128 add constraint Phone UNIQUE(Phone);  
Table altered.
```

15. Truncate the user table.

Query: truncate table user\_1128;

```
SQL> truncate table user_1128;  
Table truncated.
```

**Result:**

The DDL commands are executed successfully and shown the valid output.