```
set echo on:

prompt DROPPING THE PREVIOUSLY CREATED TABLES

DROP TABLE sungby;
DROP TABLE song;
DROP TABLE artist;
DROP TABLE album;
DROP TABLE studio;
DROP TABLE musician;

prompt CREATING THE TABLES

CREATE TABLE musician(
mid VARCHAR(6) PRIMARY KEY,
mname VARCHAR(20),
birthplace VARCHAR(25));

DESC musician;

CREATE TABLE studio(
stname VARCHAR(20) PRIMARY KEY,
staddr VARCHAR(25),
stdphn NUMBER(14));

DESC studio;

CREATE TABLE album(
alname VARCHAR(20),
alid VARCHAR(6) PRIMARY KEY,
release_yr DATE CHECK(EXTRACT(year from release_yr) >= 1945),
no_of_tracks NUMBER(3) NOT NULL,
stname VARCHAR(20) REFERENCES studio(stname),
genre VARCHAR(4) CHECK(genre IN('CAR', 'DIV', 'MOV', 'POP')),
mid VARCHAR(6) REFERENCES musician(mid));

DESC album;

CREATE TABLE artist(
arid VARCHAR(6) PRIMARY KEY,
arname VARCHAR(20),
CONSTRAINT uniq_aname UNIQUE(arname));

DESC artist;

CREATE TABLE song(
alid VARCHAR(6),
track_no VARCHAR(6),
sname VARCHAR(20),
length NUMBER(3),
genre VARCHAR(4),
PRIMARY KEY(alid, track_no),
CONSTRAINT fk_alid FOREIGN KEY(alid) REFERENCES album(alid),
CONSTRAINT chk_genre CHECK(genre IN('PHI', 'REL', 'LOV', 'DEV', 'PAT')),
CONSTRAINT chk_len CHECK(length>7 OR genre<>'PAT'));

DESC song;

CREATE TABLE sungby(
alid VARCHAR(6),
track_no VARCHAR(6),
arid VARCHAR(6) REFERENCES artist(arid),
recording_date DATE,
PRIMARY KEY(alid, track_no, arid),
CONSTRAINT fk_sungby FOREIGN KEY(alid, track_no) REFERENCES song(alid,
track_no));

DESC sungby;

prompt DESCRIBING THE TABLES

DESC musician;
DESC studio;
DESC album;
DESC artist;
DESC song;
DESC sungby;

prompt INSERTING VALUES INTO THE TABLES

INSERT INTO musician VALUES('m03', 'calvin', 'usa');
INSERT INTO musician VALUES('m01', 'miguel', 'mexico');
INSERT INTO musician VALUES('m02', 'elaine', 'france');

INSERT INTO studio VALUES('big machine', '122 pinwheel road, texas',
7445578787);
INSERT INTO studio VALUES('yg', 'seoul circle, s.korea', 4445578787);
INSERT INTO studio VALUES('sm', 'shibuya, tokyo, japan', 3445578787);

INSERT INTO album VALUES('square up', 'al01', '27-jun-2019', 4, 'yg', 'POP',
'm03');
INSERT INTO album VALUES('lionheart', 'al11', '23-may-2017', 10, 'sm', 'POP',
'm02');
INSERT INTO album VALUES('red', 'al04', '12-nov-2012', 13, 'big machine',
'MOV', 'm03');
INSERT INTO album VALUES('1989', 'al05', '2-oct-2015', 13, 'big machine',
'POP', 'm01');

INSERT INTO artist VALUES('a01', 'taylor');
INSERT INTO artist VALUES('a099', 'jennie');
INSERT INTO artist VALUES('a07', 'yuri');

INSERT INTO song VALUES('al04', 't01', '22', 212, 'PHI');
INSERT INTO song VALUES('al05', 't01', 'blank space', 221, 'LOV');
INSERT INTO song VALUES('al04', 't04', 'safe', 271, 'PAT');

INSERT INTO sungby VALUES('al05', 't01', 'a01', '27-sep-2014');
INSERT INTO sungby VALUES('al04', 't04', 'a01', '23-aug-2013');
INSERT INTO sungby VALUES('al04', 't01', 'a01', '9-may-2009');

prompt DISPLAYING THE TABLE CONTENTS

SELECT * FROM musician;
SELECT * FROM studio;
SELECT * FROM album;
SELECT * FROM artist;
SELECT * FROM song;
SELECT * FROM sungby;

prompt 1)The genre for Album can be generally categorized as CAR for Carnatic,
DIV for Divine, MOV for Movies, POP for Pop songs.

SELECT * FROM album;
INSERT INTO album VALUES('square up', 'al01', '01-dec-2019', 4, 'yg', 'RAP',
'm03');

prompt 2)The genre for Song can be PHI for philosophical, REL for
relationship, LOV for duet, DEV for devotional, PAT for patriotic type of
```

```
songs.

SELECT * FROM song;
INSERT INTO song VALUES('al01', 't04', 'solo', 233, 'JAZZ');

prompt 3)The artist ID, album ID, musician ID, and track number, studio name
are used toretrieve tuple(s) individually from respective relations.

SELECT * FROM artist;
INSERT INTO artist VALUES('a01', 'mark');
SELECT * FROM album;
INSERT INTO album VALUES('bigbang', 'al01', 2019, 4, 'yg', 'POP', 'm03');
SELECT * FROM musician;
INSERT INTO musician VALUES('m03', 'canary', 'usa');
SELECT * FROM sungby;
INSERT INTO sungby VALUES('al04', 't04', 'a01', '2-dec-2018');
SELECT * FROM studio;
INSERT INTO studio VALUES('sm', 'big ben, london, uk', 3445578787);

prompt 6)It was learnt that the artists do not have the same name.

SELECT * FROM artist;
INSERT INTO artist VALUES('a23', 'taylor');

prompt 7)The number of tracks in an album must always be recorded.

SELECT * FROM album;
INSERT INTO album VALUES('square up', 'al01', '01-dec-2019', NULL, 'yg',
'POP', 'm03');

prompt 8)The length of each song must be greater than 7 for PAT songs.

SELECT * FROM song;
INSERT INTO song VALUES('al04', 't04', 'safe', 5, 'PAT');

prompt 9)The year of release of an album can not be earlier than 1945.

SELECT * FROM album;
INSERT INTO album VALUES('green', 'al05', '8-feb-1922', 13, 'big machine',
'POP', 'm01');

prompt 10)It is necessary to represent the gender of an artist in the table.

DESC artist;
ALTER TABLE artist ADD gender VARCHAR(20);
DESC artist;

prompt 12)The phone number of each studio should be different.

ALTER TABLE studio ADD CONSTRAINT uniq_phn UNIQUE(stdphn);
DESC studio;
SELECT * FROM studio;
INSERT INTO studio VALUES('sm', 'shibuya, tokyo, japan', 4445578787);

prompt 13)An artist who sings a song for a particular track of an album can
not be recorded without the record_date.

ALTER TABLE sungby MODIFY recording_date DATE NOT NULL;
DESC sungby;
SELECT * FROM sungby;
INSERT INTO sungby VALUES('al04', 't01', 'a01', NULL);

prompt 14)It was decided to include the genre NAT for nature songs.
```

```
SELECT * FROM song;
ALTER TABLE song DROP CONSTRAINT chk_genre;
ALTER TABLE song ADD CONSTRAINT chk_genre CHECK(genre IN('PHI', 'REL', 'LOV',
'DEV', 'PAT', 'NAT'));
INSERT INTO song VALUES('al01', 't04', 'solo', 233, 'NAT');
SELECT * FROM song;

prompt 15)Due to typo-error, there may be a possibility of false information.
REM:Hence while deleting the song information, make sure that all the
corresponding information are also deleted.

ALTER TABLE sungby DROP CONSTRAINT fk_sungby;
ALTER TABLE sungby ADD CONSTRAINT fk_sungby FOREIGN KEY(alid, track_no)
REFERENCES song(alid, track_no) ON DELETE CASCADE;

SELECT * FROM song;
SELECT * FROM sungby;
DELETE FROM song WHERE sname='safe';
SELECT * FROM song;
SELECT * FROM sungby;

SELECT * FROM musician;
SELECT * FROM studio;
SELECT * FROM album;
SELECT * FROM artist;
SELECT * FROM song;
SELECT * FROM sungby;
```