**SAMIA MAZUMDER**

**CS 331**

**PROJECT #3**

CREATE TABLE ASTRO(

ASTRO\_ID NUMBER (8) PRIMARY KEY ,

ASTRO\_FNAME VARCHAR (25) NOT NULL,

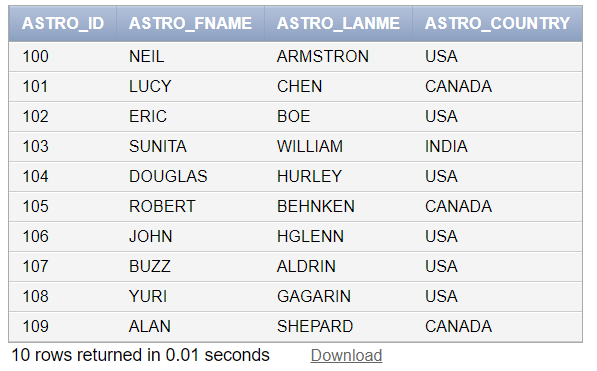
ASTRO\_LANME VARCHAR (30) NOT NULL,

ASTRO\_COUNTRY VARCHAR (6) NOT NULL

);

//SAMPLE INSERT

INSERT INTO ASTRO VALUES (100, 'NEIL', 'ARMSTRON', 'USA');



CREATE TABLE MISSION(

MISSION\_ID NUMBER (8) PRIMARY KEY ,

MISSION\_NAME VARCHAR (35) NOT NULL,

MISSION\_SDATE DATE,

MISSION\_EDATE DATE,

TARGET VARCHAR (25),

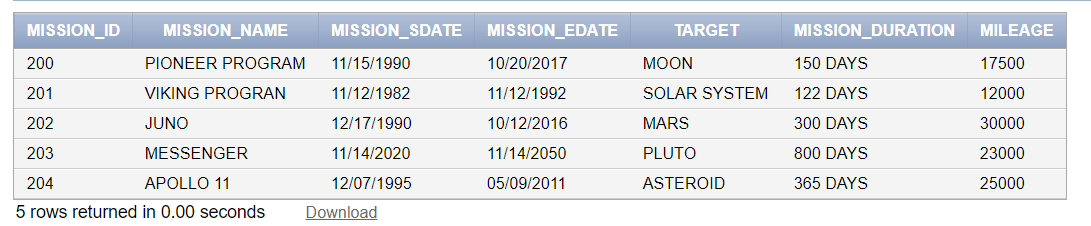
MISSION\_DURATION VARCHAR (20),

MILEAGE NUMBER (20) NOT NULL

);

//INSERT SAMPLE:

INSERT INTO MISSION VALUES (202, 'JUNO', '12/17/1990', '10/12/2016', 'MARS','300 DAYS', 30000);



//ASTR\_MISSION

CREATE TABLE ASTRO\_MISSION(

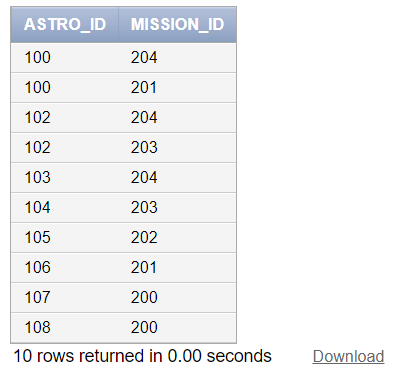
ASTRO\_ID NUMBER (8) NOT NULL,

MISSION\_ID NUMBER (8) NOT NULL,

FOREIGN KEY (MISSION\_ID) REFERENCES MISSION (MISSION\_ID),

FOREIGN KEY (ASTRO\_ID) REFERENCES ASTRO (ASTRO\_ID)

);

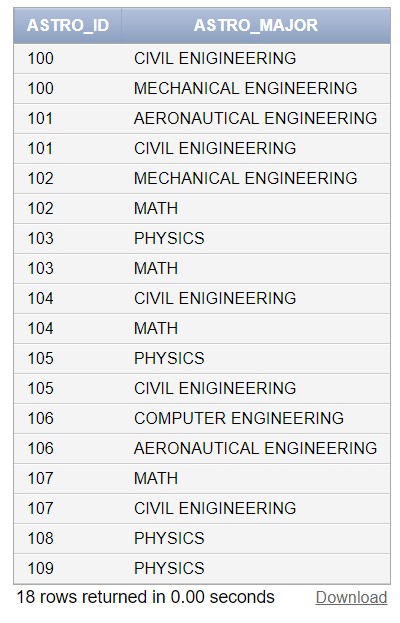


CREATE TABLE ASTRO\_MAJOR(

ASTRO\_ID NUMBER (8) ,

ASTRO\_MAJOR VARCHAR (30) NOT NULL

);



alter table "ASTRO\_MAJOR" add constraint

"ASTRO\_MAJOR\_CON" foreign key ("ASTRO\_ID") references "ASTRO" ("ASTRO\_ID")

/

//SSHIP

CREATE TABLE SSHIP(

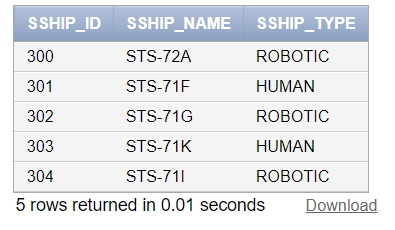
SSHIP\_ID NUMBER (8) PRIMARY KEY ,

SSHIP\_NAME VARCHAR (35) NOT NULL,

SSHIP\_TYPE VARCHAR (35) NOT NULL

);

INSERT INTO SSHIP VALUES (300, 'STS-72A','ROBOTIC');



//OBJECTIVE

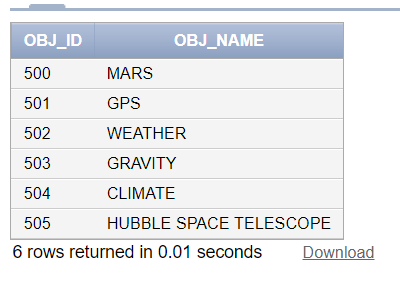
CREATE TABLE OBJECTIVE(

OBJ\_ID NUMBER (8) PRIMARY KEY ,

OBJ\_NAME VARCHAR (35) NOT NULL

);

INSERT INTO OBJECTIVE VALUES (500, 'MARS');



//MISSION\_OBJ

CREATE TABLE MISSION\_OBJ(

MISSION\_ID NUMBER (8) ,

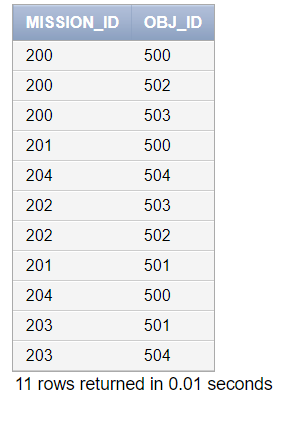
OBJ\_ID NUMBER (8),

FOREIGN KEY (MISSION\_ID) REFERENCES MISSION (MISSION\_ID),

FOREIGN KEY (OBJ\_ID) REFERENCES OBJECTIVE (OBJ\_ID)

);

INSERT INTO MISSION\_SSHIP VALUES (200,500);



//MISSION\_SSHIP

CREATE TABLE MISSION\_SSHIP(

MISSION\_ID NUMBER (8) ,

SSHIP\_ID NUMBER (8),

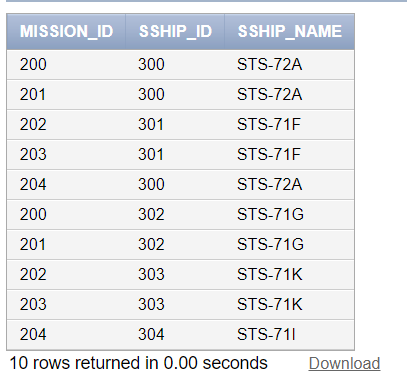
SSHIP\_NAME VARCHAR (35),

FOREIGN KEY (MISSION\_ID) REFERENCES MISSION (MISSION\_ID),

FOREIGN KEY (SSHIP\_ID) REFERENCES SSHIP (SSHIP\_ID)

);

INSERT INTO MISSION\_SSHIP VALUES (200,300, 'STS-72A');



//LSITE

CREATE TABLE LSITE(

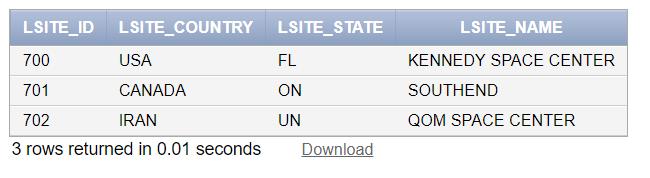
LSITE\_ID NUMBER (8) PRIMARY KEY ,

LSITE\_COUNTRY VARCHAR (25) NOT NULL,

LSITE\_STATE VARCHAR (8) NOT NULL,

LSITE\_NAME VARCHAR (40)

);



//LAUNCH

CREATE TABLE LAUNCH(

LAUNCH\_ID NUMBER (8) PRIMARY KEY ,

MISSION\_ID NUMBER (8) NOT NULL,

SSHIP\_ID NUMBER (8) NOT NULL,

LSITE\_ID NUMBER(8) NOT NULL,

LDATE DATE,

FOREIGN KEY (MISSION\_ID) REFERENCES MISSION ( MISSION\_ID),

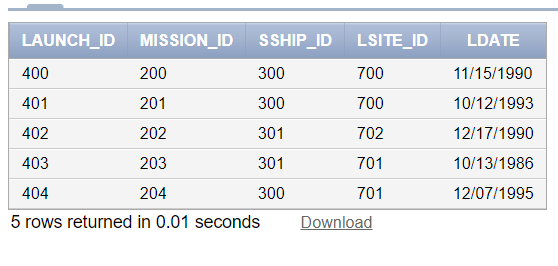
FOREIGN KEY (SSHIP\_ID) REFERENCES SSHIP(SSHIP\_ID),

FOREIGN KEY (LSITE\_ID) REFERENCES LSITE (LSITE\_ID)

);

//INSERT

INSERT INTO LAUNCH VALUES( 400, 200, 300, 700, '11/15/1990');



//STATUS

CREATE TABLE STATUS (

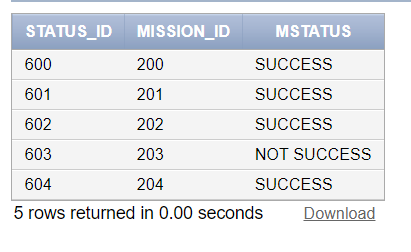
STATUS\_ID NUMBER (8) PRIMARY KEY,

MISSION\_ID NUMBER NOT NULL,

MSTATUS VARCHAR (25) NOT NULL,

FOREIGN KEY (MISSION\_ID) REFERENCES MISSION (MISSION\_ID));

INSERT INTO STATUS VALUES (600, 200,'SUCCESS');



**1)Identify all Space Shuttle missions by ‘STS-72A’. Display the mission start date, end date, launch site, landing site and mission objectives.**

SELECT M.MISSION\_SDATE, M.MISSION\_EDATE, M.TARGET, O.OBJ\_NAME, LS.LSITE\_NAME

FROM MISSION M, SSHIP SS, LSITE LS, OBJECTIVE O, LAUNCH L, MISSION\_OBJ MO

WHERE SS.SSHIP\_NAME='STS-72A'

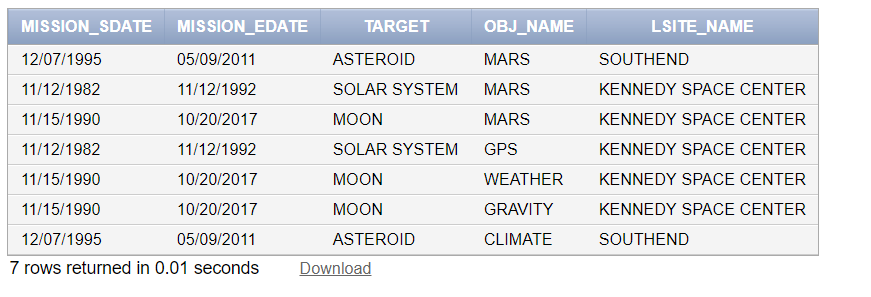
AND SS.SSHIP\_ID= L.SSHIP\_ID

AND L.MISSION\_ID=M.MISSION\_ID

AND L.LSITE\_ID=LS.LSITE\_ID

AND M.MISSION\_ID=MO.MISSION\_ID

AND MO.OBJ\_ID=O.OBJ\_ID;



**2)Identify missions by astronaut. Display three columns: Astronaut name, number of missions and total mileage. Display one row for each astronaut. Display names in alphabetical order.**

SELECT A.ASTRO\_FNAME, COUNT (M.MISSION\_ID) AS "MISSIONS", SUM (M.MILEAGE) AS "TOTAL MILEAGE"

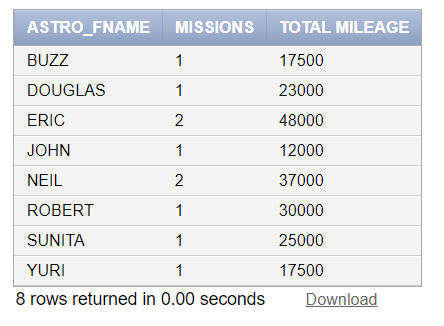
FROM ASTRO A, ASTRO\_MISSION AM, MISSION M

WHERE A.ASTRO\_ID=AM. ASTRO\_ID

AND M.MISSION\_ID= AM.MISSION\_ID

GROUP BY A.ASTRO\_FNAME

ORDER BY A.ASTRO\_FNAME;



**3)Identify all missions to ‘MOON’ launched from “SOUTHEND” in the last 20 years Display the spaceship name, start and end date of mission and objectives. Order the output by date.**

SELECT S.SSHIP\_NAME, M.MISSION\_SDATE, M.MISSION\_EDATE, O.OBJ\_NAME

FROM SSHIP S, MISSION M, OBJECTIVE O, MISSION\_OBJ MO, MISSION\_SSHIP MS, LSITE LS, LAUNCH L

WHERE S.SSHIP\_ID= MS.SSHIP\_ID

AND M.MISSION\_ID= MS.MISSION\_ID

AND MO.MISSION\_ID = M.MISSION\_ID

AND O.OBJ\_ID= MO.OBJ\_ID

AND L.SSHIP\_ID= S.SSHIP\_ID

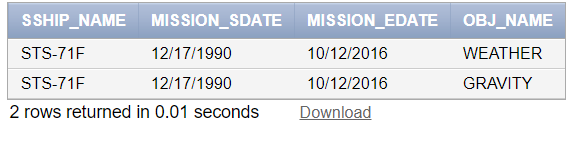
AND L.LSITE\_ID= LS.LSITE\_ID

AND M.TARGET= 'MARS'

AND LS.LSITE\_NAME= 'SOUTHEND'

AND M.MISSION\_SDATE >= '12/13/1986'

ORDER BY M.MISSION\_SDATE;



**4)Identify the missions by launch site since 2000. Display four columns: Launch site name, site country, site state and number of missions. Display one row for each launch site.**

SELECT LSITE\_NAME, LSITE\_COUNTRY, LSITE\_STATE, COUNT(M.MISSION\_ID) AS "MISSIONS"

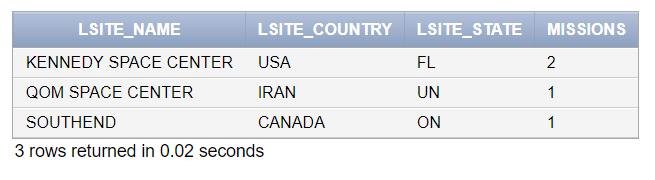
FROM MISSION M, LAUNCH L, LSITE LS

WHERE M.MISSION\_ID= L.MISSION\_ID

AND L.LSITE\_ID= LS.LSITE\_ID

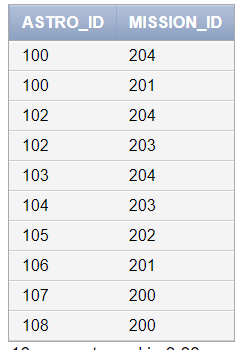
AND LDATE >='01/01/1990'

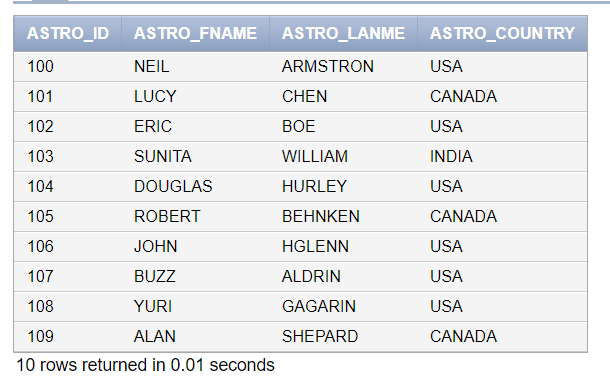
GROUP BY LS.LSITE\_NAME, LS.LSITE\_COUNTRY, LS.LSITE\_STATE;



**5) Reassign astronaut Sally Ride from Space Shuttle mission STS-7 to STS-8. Identify the SQL required to implement.**

//BEFORE UPDATE:





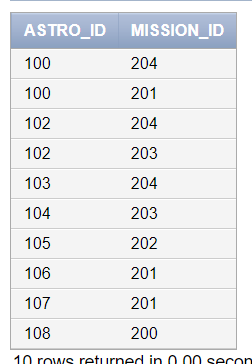
UPDATE ASTRO\_MISSION

SET MISSION\_ID= 201

WHERE ASTRO\_ID IN (SELECT ASTRO\_ID FROM ASTRO

WHERE ASTRO\_FNAME= 'BUZZ' AND ASTRO\_LANME= 'ALDRIN');

//AFTER UPDATE:



**6)Identify astronauts who have a PHYSICS degree. Use a nested select to answer this question**

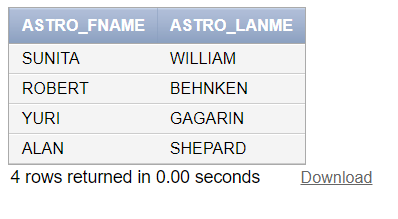
SELECT ASTRO\_FNAME, ASTRO\_LANME

FROM ASTRO

WHERE ASTRO\_ID IN (SELECT ASTRO\_ID

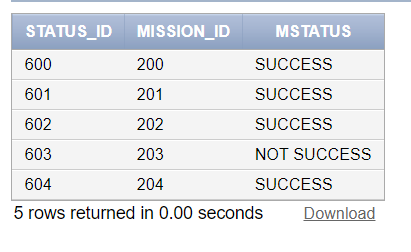
FROM ASTRO\_MAJOR

WHERE ASTRO\_MAJOR ='PHYSICS')



**7)Cancel Space Shuttle mission STS-71F. Identify the SQL to implement.**

//BEFORE

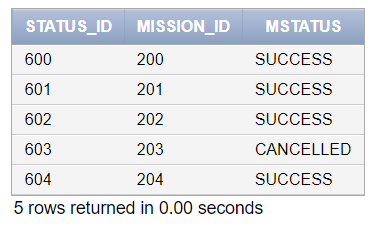


UPDATE STATUS

SET MSTATUS='CANCELLED'

WHERE MISSION\_ID= 203;

//AFTER



**8)Identify astronauts without missions in the last 10 years. Display the astronaut name and country. Use a nested select to answer this question.**

SELECT ASTRO\_FNAME, ASTRO\_LANME, ASTRO\_COUNTRY

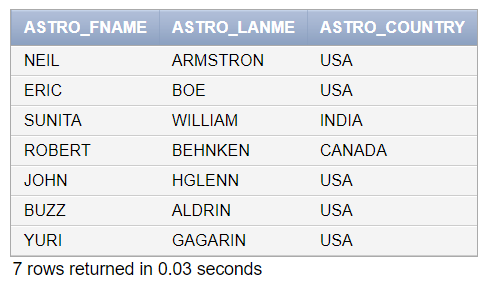
FROM ASTRO

WHERE ASTRO\_ID IN (SELECT ASTRO\_ID FROM ASTRO\_MISSION

WHERE MISSION\_ID NOT IN(SELECT MISSION\_ID

FROM MISSION

WHERE MISSION\_SDATE>='01/01/2008'));



**9) In one SQL window, reassign Astronaut A to mission 1. Don’t commit. In another SQL window,. Don’t commit. Explain your results. Resolve the problem. Disable the auto commit flag at the top of the window before performing this operation**

//UPDATE*(LEFT WINDOW)* COMMANDS:

UPDATE ASTRO\_MISSION

SET ASTRO\_ID = 106

WHERE MISSION\_ID =202;

//BUT DISABLED THE “AUTO-COMMIT” CHECKBOX.

NOW OPENED ANOTHER SQL USER SIGN IN WHICH IS *(RIGHT WINDOW)*

AND USED SAME COMMAND IN BOTH WINDOW

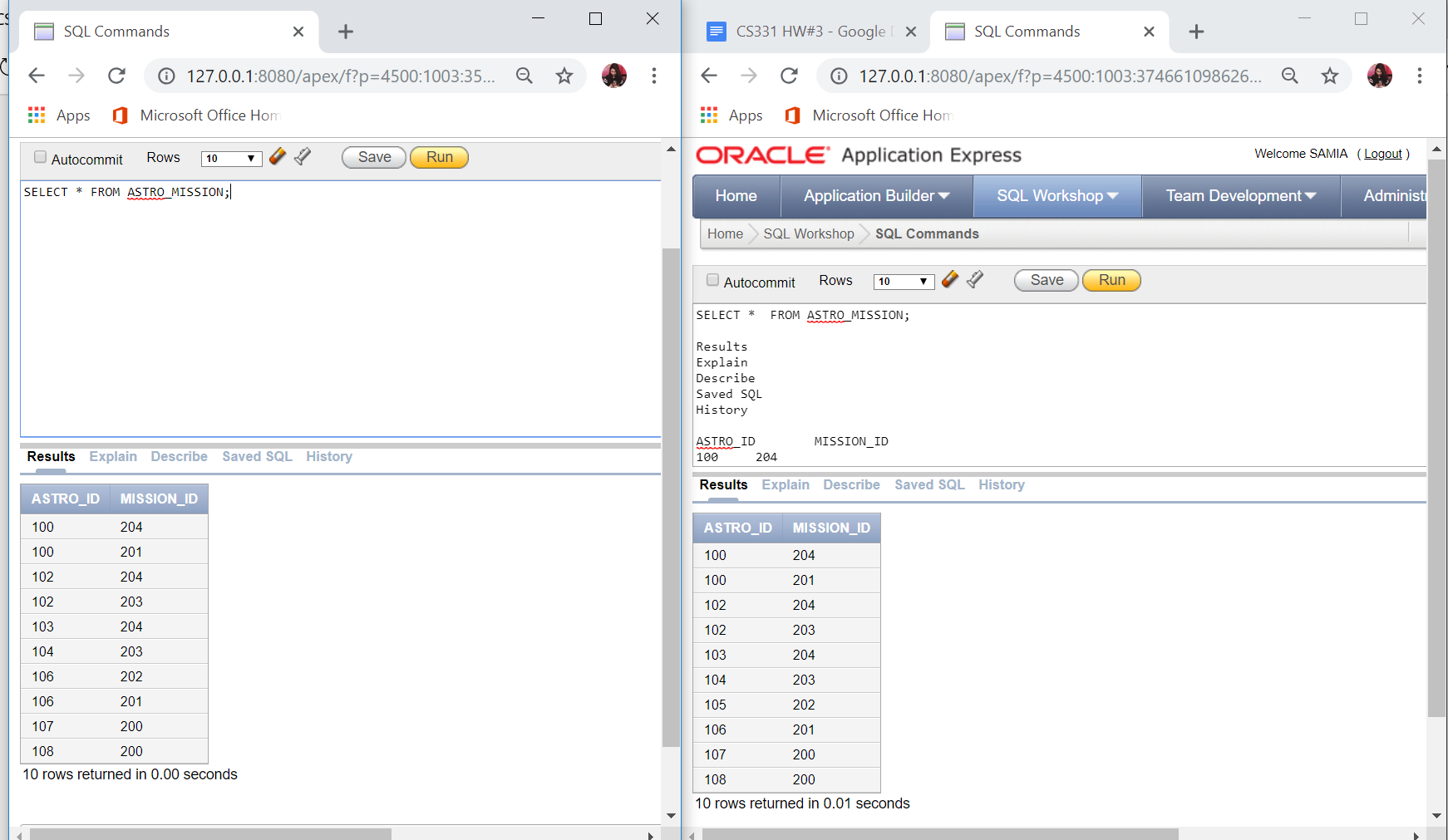
THE COMMAND IS :

SELECT \* FROM ASTRO\_MISSION;

//RESULT

AS THE WINDOW 1 THE COMMAND IS NOT COMMET OR ROLLBACK THAT WHY WINDOW 2 THE UPDATED VALUES ARE NOT SHOWING.

IF IT WAS COMMETED THEN WINDOW 2 WOULD BE ABLE TO SEE THE UPDATE VALUES.



**10). In one SQL window, delete the astronaut YURI GAGARIN. Don’t commit. In another SQL window, assign astronaut Sally Ride to a new mission. Don’t commit. Explain your results. Resolve the problem. Create a backup table before implementing. To create a backup table, enter CREATE TABLE <NEWTABLE> AS SELECT \* FROM <ORIGINALTABLE>; COMMIT; Then you can rename a table using the RENAME TABLE commit. Disable the auto commit flag at the top of the window before performing this operation.**

//COMMANDS (*LEFT WINDOW*):

DELETE FROM ASTRO\_MISSION

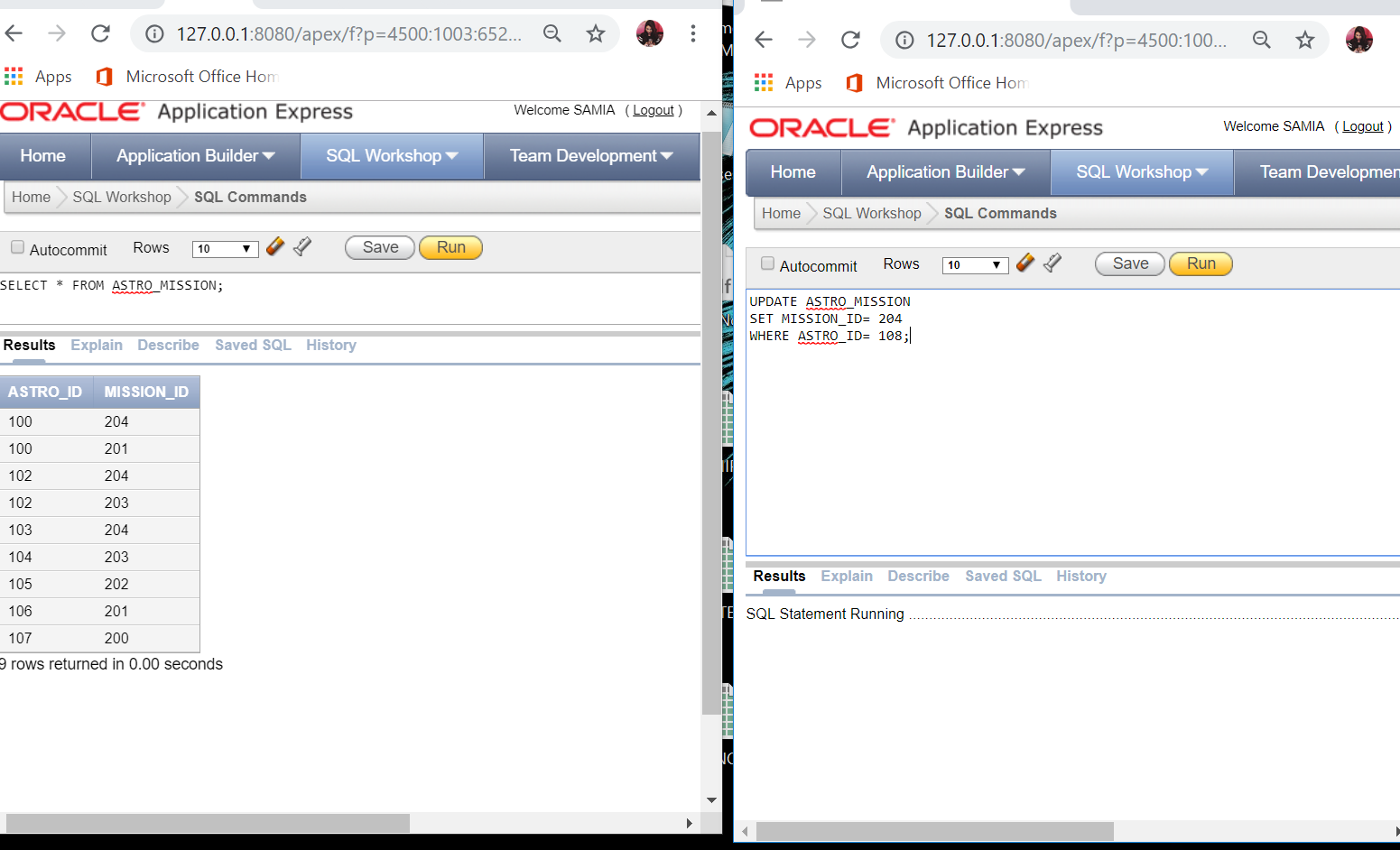
WHERE ASTRO\_ID= 108;

// COMMANDS*(RIGHT WINDOW*):

UPDATE ASTRO\_MISSION

SET MISSION\_ID= 204

WHERE ASTRO\_ID= 108;



SO LEFT WINDOW I HAVE DELETED THE ASTRONAUT NAMED YURI GAGARIN AND ASTRO\_ID =108 AND RIGHT WINDOW HAVE ASSIGNED SAME ASTRONAUT ASTR\_ID=108 FROM MISSION \_ID=200 TO MISSION\_ID =204 AND ITS GIVING “SQL Statement Running…………..” MEANING THERE IS ANOTHER WINDOW OPEN AND THERE IS A LOCK IN THE TABLE WHICH IS THE HIGHEST LOCK. TO RESOLVE THIS

PROBLEM IS TO FINISH THE RUNNING SESSION ON THE ONE WINDOW TO WORK IN THE OTHER WINDOW OR BY USING COMMAND ROLLBACK OR COMMIT CAN RELEASE THE LOCK ON THE TABLE ASTR\_MISSION.

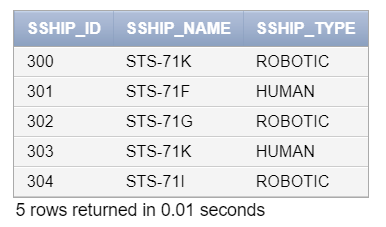
**11) In one SQL window, rename the Space Shuttle STS-72A to STS-71K. Don’t commit. In another SQL window, change the Space Shuttle STS-72A to STS-71G. Don’t commit. Quit both Oracle sessions. Login to Oracle and display all information for the Space Shuttle Enterprise.** Explain your results. Disable the auto commit flag at the top of the windows before performing this operation.

*//*COMMANDS *( LEFT WINDOW)*

UPDATE SSHIP

SET SSHIP\_NAME='STS-71K'

WHERE SSHIP\_NAME='STS-72A';

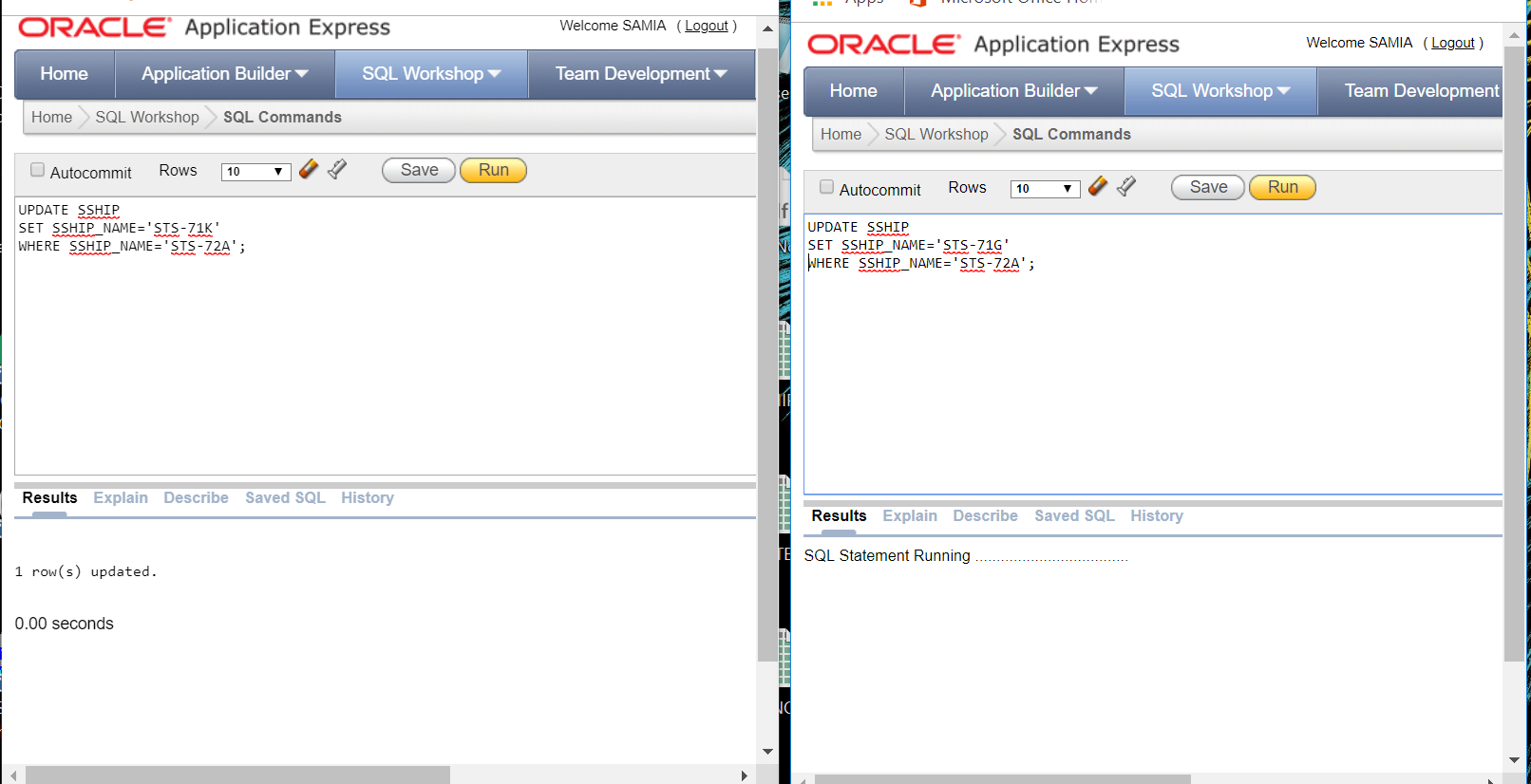


//COMMANDS *(RIGHT WINDOW)*

UPDATE SSHIP

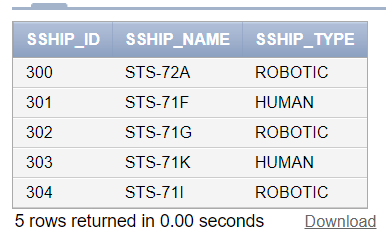
SET SSHIP\_NAME='STS-71G'

WHERE SSHIP\_NAME='STS-72A';



IN HERE WE CAN SEE THAT LEFT SIDE COMMAND WAS SUCCESSFULLY WORKED AND UPDATED THE ROW BUT ON THE RIGHT WINDOW IT HAS THE “SQL Statement Running……..” MESSAGE THAT THE SSHIP TABLE HAS A LOCK ON IT AND ONLY WAY TO RESOLVE IT IS TO USE COMMAND ‘COMMIT OR ROLLBACK’ OR FINISH THE SESSION THATS RUNNING IN THIS CASE THE SESSION IS RUNNING IN THE LEFT WINDOW.

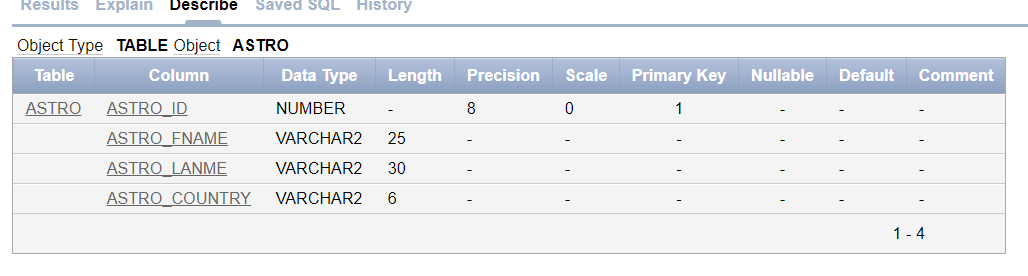
//NOW AFTER QUITTING FROM BOTH WINDOW AGAIN LOGGING INTO ORACLE:



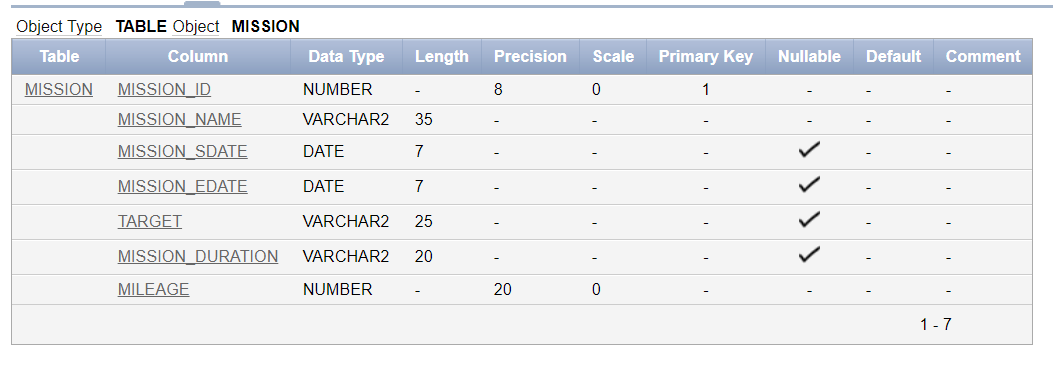
AS WE CAN SEE NOTHING HAS UPDATED AS I DIDN’T USE THE COMMIT COMMAND IN SSHIP TABLE.

**12)table desc**

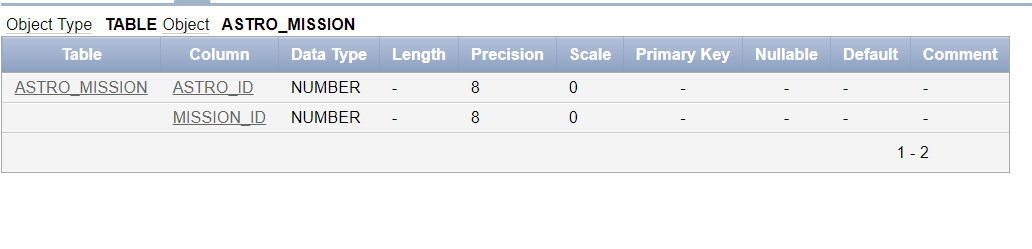
**//astro**

****

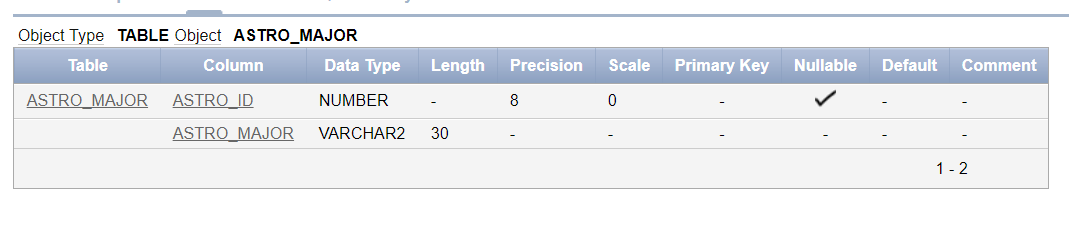
**//Mission**

****

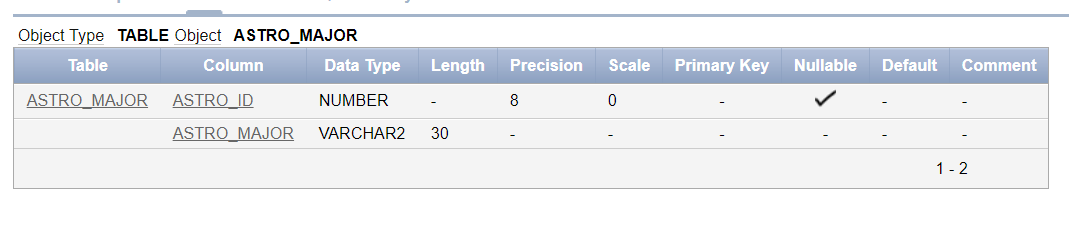
**//astro\_mission**

****

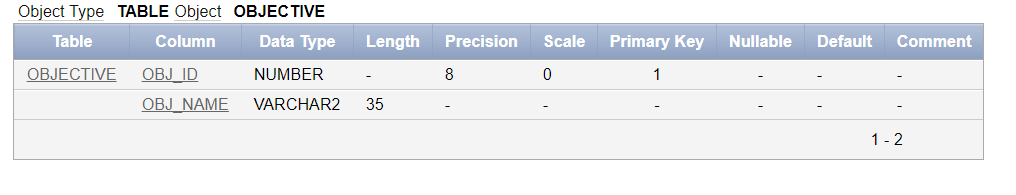
**//astro\_major**

****

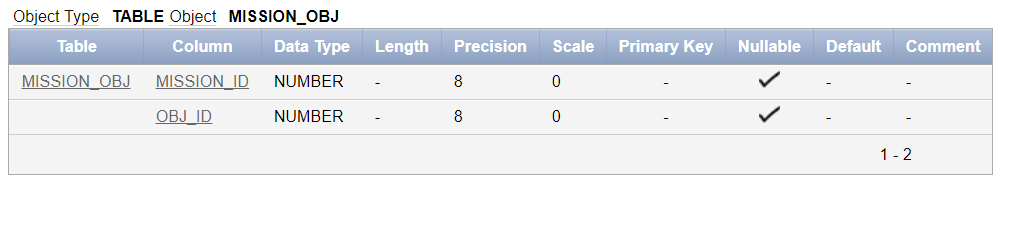
**//sship**

****

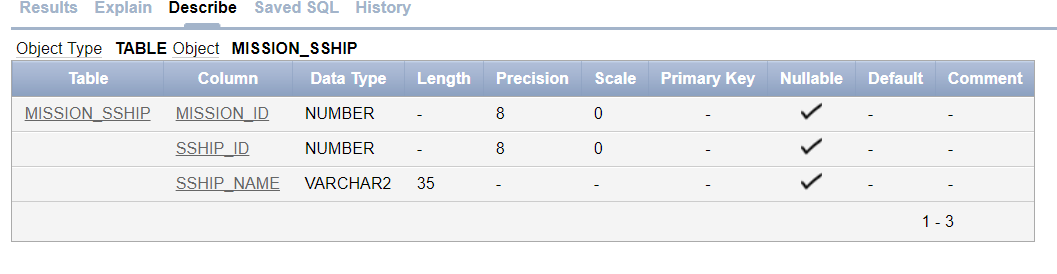
**//objective**

****

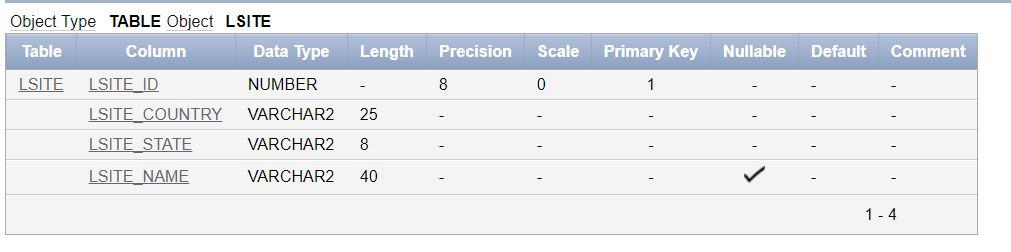
**//mission\_obj**

****

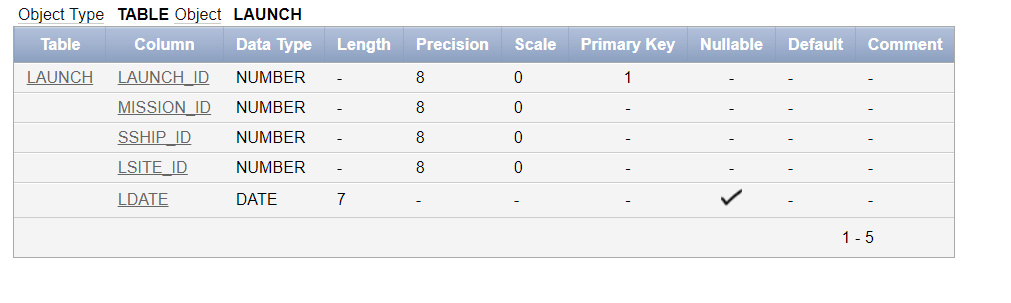
**//mission\_sship**

****

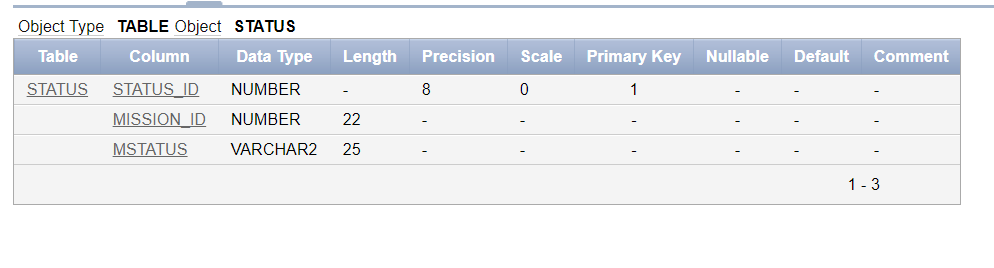
**//lsite**

****

**//launch**

****

**//status**

****

**13) Display the Oracle version by entering**

