

DBMS ASSIGNMENTS

PART A

INDEX

<u>SER</u>	<u>DBMS EXERCISES (PART A)</u>	<u>PAGE NO</u>
<u>1</u>	ACTIVITY – 1: DATABASE : STUDENT (DDL, DML STATEMENTS)	<u>03 -06</u>
<u>2</u>	ACTIVITY 2: (SELECT CLAUSE, ARITHMETIC OPERATORS)	<u>07-14</u>
<u>3</u>	ACTIVITY 3: (LOGICAL, RELATIONAL OPERATORS)	<u>15-21</u>
<u>4</u>	ACTIVITY 4: (DATE FUNCTIONS)	<u>22-27</u>
<u>5</u>	ACTIVITY 5: (NUMERIC, CHARACTER FUNCTIONS)	<u>28-33</u>
<u>6</u>	ACTIVITY : 6 (SET OPERATORS)	<u>34-39</u>
<u>7</u>	ACTIVITY 7: (VIEWS) DATABASE: RAILWAY RESERVATION SYSTEM	<u>40-47</u>

DBMS ASSIGNMENTS

PART A

Activity 1:

Database : Student (DDL, DML statements)

Table: Student

Name	Regno	Class	Major
Smith	17	1	CS
Brown	8	2	CS

Table: Course

CourseName	CourseNumber	CreditHours	Department
Intro to computer science	CS1310	4	CS
Data Structure	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

Table: Section

SectionIdentifier	CourseNumber	Year	Instructor
85	MATH2410	98	King
92	CS1310	98	Andreson
102	CS3320	99	Knuth
112	MATH2410	99	Chang
119	CS1310	99	Andreson
135	CS3380	99	Stone

Table: Grade_report

Regno	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

1. CREATE TABLES USING CREATE STATEMENT
2. INSERT ROWS TO INDIVIDUAL TABLES USING INSERT STATEMENT
3. ALTER TABLE SECTION ADD NEW FIELD SECTION AND UPDATE THE RECORDS
4. DELETE BROWN'S GRADE REPORT
5. DROP THE TABLE SECTION

1. CREATE TABLES USING CREATE STATEMENT

```
Run SQL Command Line

SQL> Create table student (name varchar2(20),regno int primary key, class int,ma
jor varchar2(4));
Table created.
SQL>
```

```
Run SQL Command Line

SQL> Create table student (name varchar2(20),regno int primary key, class int,ma
jor varchar2(4));
Table created.
SQL> Create table course(coursename varchar2(20),coursenumber varchar2(10),credi
thours int,department varchar2(10));
Table created.
SQL>
```

```
Run SQL Command Line

SQL*Plus: Release 10.2.0.1.0 - Production on Sat Nov 3 11:48:58 2018
Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> connect scott/tiger
Connected.
SQL> Create table section(sectionidentifier int, coursenumber varchar2(10),year
int,instructor varchar2(15) );
Table created.
SQL> _
```

```
Run SQL Command Line

SQL*Plus: Release 10.2.0.1.0 - Production on Sat Nov 3 11:48:58 2018
Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> connect scott/tiger
Connected.
SQL> Create table section(sectionidentifier int, coursenumber varchar2(10),year
int,instructor varchar2(15) );
Table created.
SQL> Create table gradereport(regno int, sectionidentifier int, grade varchar2(1
) );
Table created.
SQL> _
```

2. INSERT ROWS TO INDIVIDUAL TABLES USING INSERT STATEMENT

```
Run SQL Command Line

SQL> Insert into student values('Smith',17,1,'CS');
1 row created.

SQL> Insert into student values('Brown',8,s,'CS');
Insert into student values('Brown',8,s,'CS')
*
ERROR at line 1:
ORA-00984: column not allowed here

SQL> Insert into student values('Brown',8,2,'CS');
1 row created.

SQL> █
```

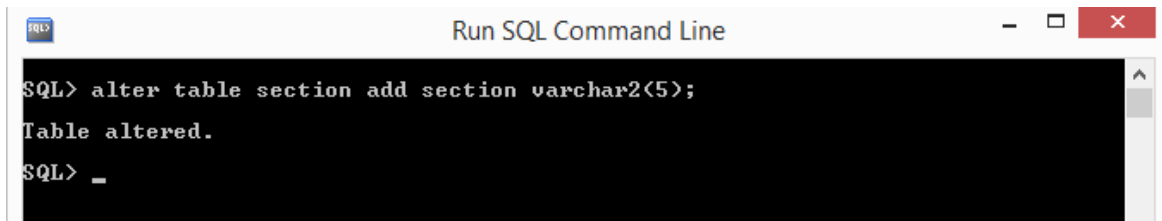
```
Run SQL Command Line

SQL> ALTER TABLE COURSE MODIFY COURSENAME VARCHAR2(30);
Table altered.

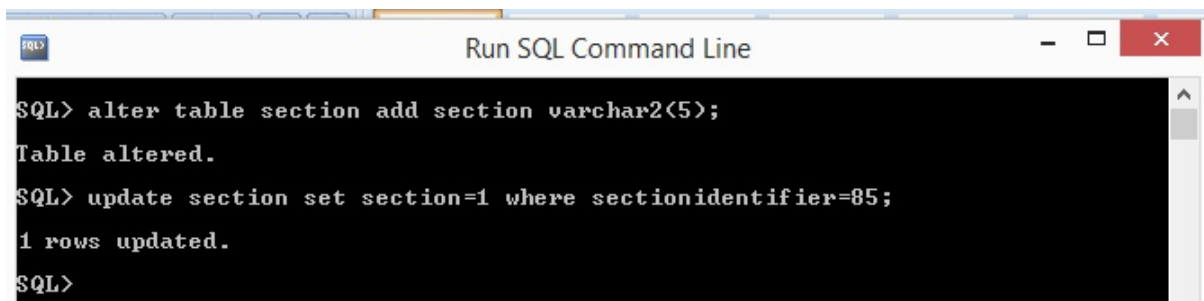
SQL> insert into course values('&name','&cno','&hr','&dept');
Enter value for name: Intro to Computer Science
Enter value for cno: CS1310
Enter value for hr: 4
Enter value for dept: CS
old 1: insert into course values('&name','&cno','&hr','&dept')
new 1: insert into course values('Intro to Computer Science','CS1310',4,'CS')
1 row created.

SQL>
```

3. ALTER TABLE SECTION ADD NEW FIELD SECTION AND UPDATE THE RECORDS

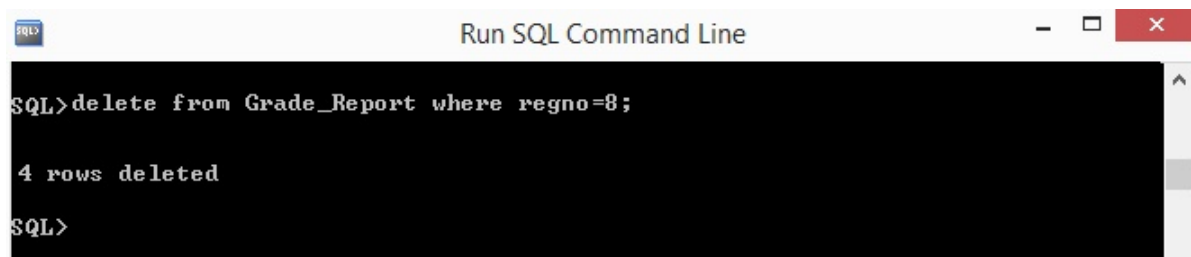


```
SQL> alter table section add section varchar2(5);
Table altered.
SQL> _
```



```
SQL> alter table section add section varchar2(5);
Table altered.
SQL> update section set section=1 where sectionidentifier=85;
1 rows updated.
SQL>
```

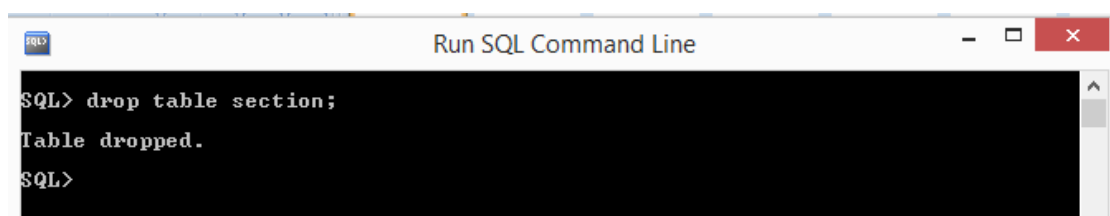
4. DELETE BROWN'S GRADE REPORT



```
SQL> delete from Grade_Report where regno=8;

4 rows deleted
SQL>
```

5. DROP THE TABLE SECTION



```
SQL> drop table section;
Table dropped.
SQL>
```

Activity 2: (Select clause, Arithmetic Operators)**Database: employee**Create Following **tables** and insert **tuples** with suitable constraints**EMPLOYEE**

EMPID	FIRSTNAME	LASTNAME	Hire Date	ADDRESS	CITY
1001	George	Smith	11-May-06	83 first street	Paris
1002	Mary	Jones	25-Feb-08	842 Vine Ave	Losantiville
1012	Sam	Tones	12-Sep-05	33 Elm St.	Paris
1015	Peter	Thompson	19-Dec-06	11 Red Road	Paris
1016	Sarath	Sharma	22-Aug-07	440 MG Road	New Delhi
1020	Monika	Gupta	07-Jun-08	9 Bandra	Mumbai

EMPSALARY

EMPID	SALARY	BENEFITS	DESIGNATION
1001	10000	3000	Manager
1002	8000	1200	Salesman
1012	20000	5000	Director
1015	6500	1300	Clerk
1016	6000	1000	Clerk
1020	8000	1200	Salesman

Write queries for the following

1. To display FIRSTNAME, LASTNAME, ADDRESS AND CITY of all employees living in PARIS.
2. To display the content of employee table in descending order of FIRSTNAME
3. Select FIRSTNAME and SALARY of salesman
4. To display the FIRSTNAME, LASTNAME, AND TOTAL SALARY of all employees from the table EMPLOYEE and EMPSALARY. Where TOTAL SALARY is calculated as SALARY+BENEFITS
5. List the Names of employees, who are more than 1 year old in the organization
6. Count number of distinct DESIGNATION from EMPSALARY
7. List the employees whose names have exactly 6 characters
8. Add new column PHONE_NO to EMPLOYEE and update the records
9. List employee names, who have joined before 15-Jun-08 and after 16-Jun-07
10. Generate Salary slip with Name, Salary, Benefits, HRA-50%, DA-30%, PF-12%, Calculate gross. Order the result in descending order of the gross.

CREATING EMPLOYEE AND EMP SAL TABLES

```
Run SQL Command Line

SQL*Plus: Release 10.2.0.1.0 - Production on Sat Nov 3 16:44:01 2018
Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> create table employee (empid int primary key,firstname varchar2(20)
  2  ,lastname varchar2(20),hire_date date,address varchar2(35),
  3  city varchar2(25));
SP2-0640: Not connected
SQL> connect scott/tiger
Connected.
SQL> ed
SP2-0110: Cannot create save file "afiedt.buf"
SQL> create table employee (empid int primary key,firstname varchar2(20)
  2  ,lastname varchar2(20),hire_date date,address varchar2(35),
  3  city varchar2(25));

Table created.

SQL>
```

```
Run SQL Command Line

SQL> create table empsal (empid int, salary float,
  2  benefits float, designation varchar2(25), foreign key(empid)
  3  References employee(empid))
  4  ;

Table created.

SQL> _
```

INSERTING RECORDS IN EMPLOYEE TABLE

```
Run SQL Command Line

SQL>
SQL>
SQL>
SQL>
SQL>
SQL> insert into employee values(&empid,&fname,&lname,&date',
  2  '&addr','&city');
Enter value for empid: 1001
Enter value for fname: George
Enter value for lname: Smith
Enter value for date: 11-May-06
old 1: insert into employee values(&empid,&fname,&lname,&date',
new 1: insert into employee values(1001,'George','Smith','11-May-06',
Enter value for addr: 83 First Street
Enter value for city: paris
old 2: '&addr','&city')
new 2: '83 First Street','paris')

1 row created.

SQL>
```



```
Run SQL Command Line

SQL> desc empsalary
Name                                         Null?      Type
-----
EMPID                                         NUMBER(38)
SALARY                                       FLOAT(126)
BENEFITS                                    FLOAT(126)
DESIGNATION                                VARCHAR2(25)

SQL> insert into empsalary values(&empid,&salary,&benifit,&'&design');
Enter value for empid: 1001
Enter value for salary: 10000
Enter value for benifit: 3000
Enter value for design: Manager
old 1: insert into empsalary values(&empid,&salary,&benifit,&'&design')
new 1: insert into empsalary values(1001,10000,3000,'Manager')

1 row created.

SQL> /
Enter value for empid: _
```

```
Run SQL Command Line

Enter value for city: paris
old 2: '&addr','&city')
new 2: '83 First Street','paris')

1 row created.

SQL> /
Enter value for empid: 1002
Enter value for fname: Mary
Enter value for lname: Jones
Enter value for date: 25-feb-08
old 1: insert into employee values(&empid,&'&fname','&lname','&date',
new 1: insert into employee values(1002,'Mary','Jones','25-feb-08',
Enter value for addr: 842 Vine Ave
Enter value for city: Losantivile
old 2: '&addr','&city')
new 2: '842 Vine Ave','Losantivile')

1 row created.

SQL> /
Enter value for empid: 1012
Enter value for fname: Sam
Enter value for lname: Tones
Enter value for date:
```

1. TO DISPLAY FIRSTNAME, LASTNAME, ADDRESS AND CITY OF ALL EMPLOYEES LIVING IN PARIS.

```
Run SQL Command Line
SQL> select firstname,lastname,address,city from employee where city='Paris';
FIRSTNAME      LASTNAME      ADDRESS
-----
CITY
-----
Sam
Paris          Tones         33 Elm St
Peter
Paris          Thompson       22 Red Road
SQL> _
```

2. TO DISPLAY THE CONTENT OF EMPLOYEE TABLE IN DESCENDING ORDER OF FIRSTNAME.

```
Run
EMPID FIRSTNAME      LASTNAME      HIRE_DATE
-----
ADDRESS          CITY
-----
1016 Sarath      Sharma        22-AUG-07
440 MG road      New Delhi
1012 Sam         Tones         12-SEP-05
33 Elm St        Paris
1015 Peter       Thompson       19-DEC-06
22 Red Road      Paris
EMPID FIRSTNAME      LASTNAME      HIRE_DATE
-----
ADDRESS          CITY
-----
1020 Monika      Gupta         07-JUN-08
9 Bandra         Mumbai
1002 Mary        Jones         25-FEB-08
842 Vine Ave     Losantivile
1001 George      Smith         11-MAY-06
83 First Street  paris
6 rows selected.
SQL> _
```

3. SELECT FIRSTNAME AND SALARY OF SALESMAN

```
Run SQL Command Line

SQL> select e.firstname,s.salary from employee e, empsalary s where e.empid=s.empid;

FIRSTNAME          SALARY
-----
George             10000
Mary               8000
Sam               20000
Peter              6500
Sarath             6000
Monika            8000

6 rows selected.

SQL> _
```

4. TO DISPLAY THE FIRSTNAME, LASTNAME, AND TOTAL SALARY OF ALL EMPLOYEES FROM THE TABLE EMPLOYEE AND EMPSALARY. WHERE TOTAL SALARY IS CALCULATED AS SALARY+BENEFITS.

```
Run SQL Command Line

Sarath             6000
Monika            8000

6 rows selected.

SQL> select e.firstname,s.salary+s.benefits from employee e, empsalary s where e.empid=s.empid;

FIRSTNAME          S.SALARY+S.BENEFITS
-----
George             13000
Mary               9200
Sam               25000
Peter              7800
Sarath             7000
Monika            9200

6 rows selected.

SQL> _
```

5. LIST THE NAMES OF EMPLOYEES, WHO ARE MORE THAN 1 YEAR OLD IN THE ORGANIZATION

```
Run SQL Command Line

6 rows selected.

SQL> select firstname from employee where extract (year from hire_date)>>1;

FIRSTNAME
-----
George
Mary
Sam
Peter
Sarath
Monika

6 rows selected.

SQL> _
```

6. COUNT NUMBER OF DISTINCT DESINGATION FROM EMPSALARY

```

6 rows selected.

SQL> select firstname from employee where extract (year from hire_date)>>1;

FIRSTNAME
-----
George
Mary
Sam
Peter
Sarath
Monika

6 rows selected.

SQL> select count(designation) from empsalary;

COUNT(DESIGNATION)
-----
6

SQL> _

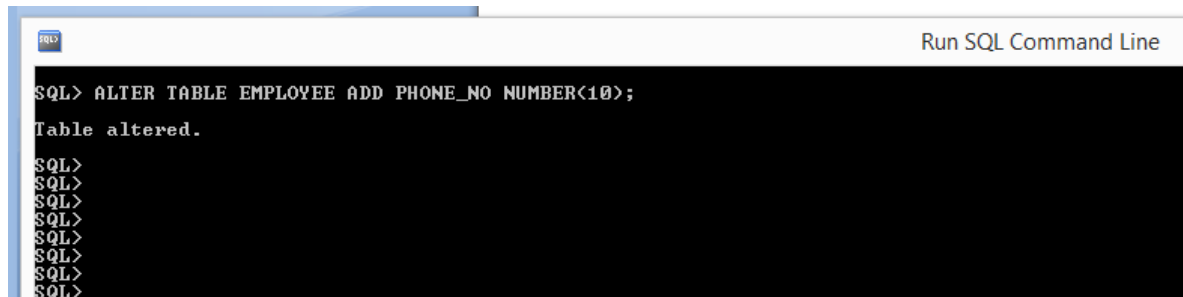
```

7. LIST THE EMPLOYEE WHOSE NAME HAS EXACTLY 6 CHARACTERS.

```
SQL> select * from employee where length(firstname)=6;
```

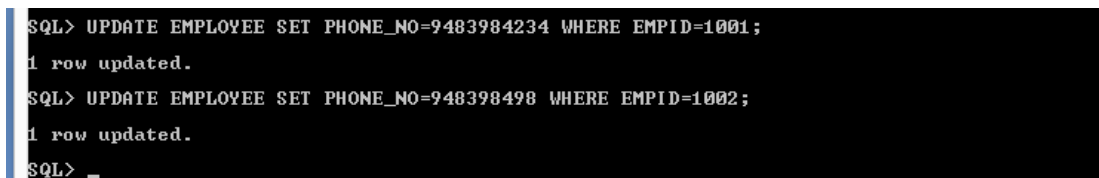
EMPID	FIRSTNAME	LASTNAME	HIRE_DATE
83	George 1001 George First Street	Smith paris	11-MAY-06
440	Sarath 1016 Sarath MG road	Sharma New Delhi	22-AUG-07
9	Monika 1020 Monika Bandra	Gupta Mumbai	07-JUN-08

8. ADD NEW COLUMN PHONE_NO TO EMPLOYEE AND UPDATE THE RECORDS.



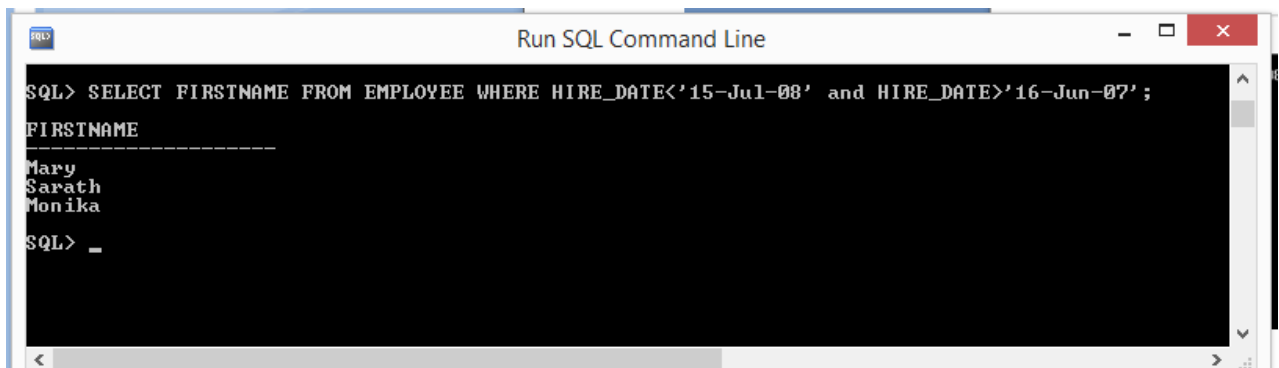
```
SQL> ALTER TABLE EMPLOYEE ADD PHONE_NO NUMBER(10);
Table altered.
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

UPDATING RECORDS:



```
SQL> UPDATE EMPLOYEE SET PHONE_NO=9483984234 WHERE EMPID=1001;
1 row updated.
SQL> UPDATE EMPLOYEE SET PHONE_NO=948398498 WHERE EMPID=1002;
1 row updated.
SQL> _
```

9. LIST EMPLOYEE NAMES, WHO HAVE JOINED BEFORE 15-JU-08 AND AFTER 16-JUN-07.



```
SQL> SELECT FIRSTNAME FROM EMPLOYEE WHERE HIRE_DATE<'15-Jul-08' and HIRE_DATE>'16-Jun-07';
FIRSTNAME
-----
Mary
Sarath
Monika
SQL> _
```

10. GENERATE SALARY SLIP WITH NAME, SALARY, BENEFITS, HRA-50%, DA-30%, PF-12%, CALCULATE GROSS, ORDER THE RESULT IN DESCENDING ORDER OF GROSS.

```

Run SQL Command Line
ORA-00904: "S"."SALARY": invalid identifier

SQL> SELECT E.FIRSTNAME,S.SALARY,S.BENEFITS,S.SALARY*50/100+S.SALARY*30/100,S.SALARY*12/100
2  +S.SALARY FROM EMPLOYEE E,EMPSALARY S WHERE E.EMPID=S.EMPID
3  ORDER BY
4  S.SALARY*50/100+S.SALARY*30/100,S.SALARY*12/100+S.SALARY;

FIRSTNAME      SALARY      BENEFITS  S.SALARY*50/100+S.SALARY*30/100
S.SALARY*12/100+S.SALARY
Sarath          6720        6000      1000      4800
Peter           7280        6500      1300      5200
Mary            8960        8000      1200      6400

FIRSTNAME      SALARY      BENEFITS  S.SALARY*50/100+S.SALARY*30/100
S.SALARY*12/100+S.SALARY
Monika          8960        8000      1200      6400
George         11200       10000      3000      8000
Sam            22400       20000      5000     16000

6 rows selected.
SQL>

```

Activity 3: (Logical, Relational Operators)

Database: Library

Create Following **tables** and insert **tuples** with suitable constraints

Table: Books

Book_Id	Book_name	Author_Name	Publishers	Price	Type	Quantity
C0001	The Klon and I	Lata Kappor	EPP	355	Novel	5
F0001	The Tears	William Hopkins	First Publ	650	Fiction	20
T0001	My First C++	Brain & Brooke	ERP	350	Text	10
T0002	C++ Brainworks	A.W.Rossaine	TDH	350	Text	15
F0002	Thunderbolts	Ana Roberts	First Publ.	750	Fiction	50

Table : Issued

Book_Id	Quantity_Issued
T0001	4
C0001	5
F0001	2
T0002	5
F0002	8

Write queries for the following

1. To show Book name, Author name and price of books of **First Publ.** publisher
2. Display Book id, Book name and publisher of books having quantity more than 8 and price less than 500
3. Select Book id, book name, author name of books which is published by other than ERP publishers and price between 300 to 700
4. Generate a Bill with Book_id, Book_name, Publisher, Price, Quantity, 4% of VAT –Total”
5. Display book details with book id's C0001, F0001, T0002, F0002 (Hint: use IN operator)
6. Display Book list other than, type Novel and Fiction
7. Display book details with author name starts with letter 'A'
8. Display book details with author name starts with letter 'T' and ends with 'S'
9. Select BookId, BookName, Author Name , Quantity Issued where Books.BookId = Issued.BookId
10. List the book_name, Author_name, Price. In ascending order of Book_name and then on descending order of price

CREATING TABLE BOOKS

```
Run SQL Command Line

SQL> CREATE TABLE BOOKS (BOOK_ID VARCHAR(10) PRIMARY KEY,
2  BOOK_NAME VARCHAR(20),
3  AUTHOR_NAME VARCHAR(20),
4  PUBLISHERS VARCHAR(20),
5  PRICE NUMBER(10,2),
6  TYPE VARCHAR(10),
7  QUANTITY NUMBER(5));

Table created.

SQL>
```

INSERTING RECORDS IN BOOKS TABLE

```
Run SQL Command Line

SQL>
SQL>
SQL>
SQL>
SQL> INSERT INTO BOOKS VALUES('&BOOKID','&BOOKNAME','&AUTHORNAME','&PUBLISHERS','&PRICE','&TYPE','&
Enter value for bookid: C001
Enter value for bookname: THE KLONE AND I
Enter value for authorname: LAA KAPPOR
Enter value for publishers: EPP
Enter value for price: 355
Enter value for type: NOVEL
Enter value for qty: 5
old 1: INSERT INTO BOOKS VALUES('&BOOKID','&BOOKNAME','&AUTHORNAME','&PUBLISHERS','&PRICE','&TYPE'
new 1: INSERT INTO BOOKS VALUES('C001','THE KLONE AND I','LAA KAPPOR','EPP',355,'NOVEL',5)

1 row created.

SQL> /
Enter value for bookid: F0001
Enter value for bookname: THE TEARS
Enter value for authorname: WILLIAM HOPKINS
Enter value for publishers: FIRST PUBL
Enter value for price: 650
Enter value for type: FICTION
Enter value for qty: 20
old 1: INSERT INTO BOOKS VALUES('&BOOKID','&BOOKNAME','&AUTHORNAME','&PUBLISHERS','&PRICE','&TYPE'
new 1: INSERT INTO BOOKS VALUES('F0001','THE TEARS','WILLIAM HOPKINS','FIRST PUBL',650,'FICTION'

1 row created.

SQL>
```

```
Run SQL Command Line

new 1: INSERT INTO BOOKS VALUES('F0001','THE TEARS','WILLIAM HOPKINS','FIRST PUBL',650,'FICTION'

1 row created.

SQL> /
Enter value for bookid: T0001
Enter value for bookname: MYFIRST C++
Enter value for authorname: BRAIN AND BROOKE
Enter value for publishers: ERP
Enter value for price: 350
Enter value for type: TEXT
Enter value for qty: 10
old 1: INSERT INTO BOOKS VALUES('&BOOKID','&BOOKNAME','&AUTHORNAME','&PUBLISHERS','&PRICE','&TYPE'
new 1: INSERT INTO BOOKS VALUES('T0001','MYFIRST C++','BRAIN AND BROOKE','ERP',350,'TEXT',10)

1 row created.

SQL> /
Enter value for bookid: T0002
Enter value for bookname: C++ BRAINWORKS
Enter value for authorname: AW ROSSAINE
Enter value for publishers: TDH
Enter value for price: 350
Enter value for type: TEXT
Enter value for qty: 15
old 1: INSERT INTO BOOKS VALUES('&BOOKID','&BOOKNAME','&AUTHORNAME','&PUBLISHERS','&PRICE','&TYPE'
new 1: INSERT INTO BOOKS VALUES('T0002','C++ BRAINWORKS','AW ROSSAINE','TDH',350,'TEXT',15)

1 row created.

SQL>
```



```
Run SQL Command Line

Enter value for qty: 15
old 1: INSERT INTO BOOKS VALUES('&BOOKID','&BOOKNAME','&AUTHORNAME','&PUBLISHERS','&PRICE','&TYP
new 1: INSERT INTO BOOKS VALUES('T0002','C++ BRAINWORKS','AW ROSSAINE','TDH',350,'TEXT',15)
1 row created.

SQL> /
Enter value for bookid: F0002
Enter value for bookname: THUNDERBOLTS
Enter value for authorname: ANA ROBERTS
Enter value for publishers: FIRST PUBL
Enter value for price: 750
Enter value for type: FICTION
Enter value for qty: 50
old 1: INSERT INTO BOOKS VALUES('&BOOKID','&BOOKNAME','&AUTHORNAME','&PUBLISHERS','&PRICE','&TYP
new 1: INSERT INTO BOOKS VALUES('F0002','THUNDERBOLTS','ANA ROBERTS','FIRST PUBL',750,'FICTION
1 row created.

SQL>
```

CREATING TABLE: ISSUED.

```
Run SQL Command Line

SQL> CREATE TABLE ISSUED (BOOK_ID VARCHAR(10) REFERENCES BOOKS(BOOK_ID),
2 QUANTITY_ISSUED NUMBER(5));
Table created.

SQL>
```

INSERTING RECORDS IN ISSUE TABLE

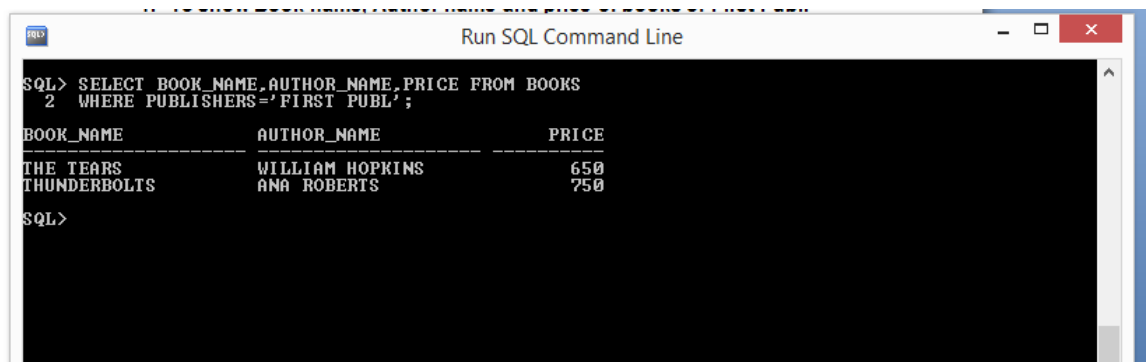
```
Run SQL Command Line

1 row created.

SQL> /
Enter value for bookid: c0001
Enter value for qty: 5
old 1: insert into issued values('&bookid',&qty)
new 1: insert into issued values('c0001',5)
insert into issued values('c0001',5)
*
ERROR at line 1:
ORA-02291: integrity constraint (SCOTT.SYS_C004171) violated - parent key not
found

SQL> /
```

1. TO SHOW BOOK NAME, AUTHOR NAME AND PRICE OF BOOKS OF FIRST PUBL. PUBLISHER

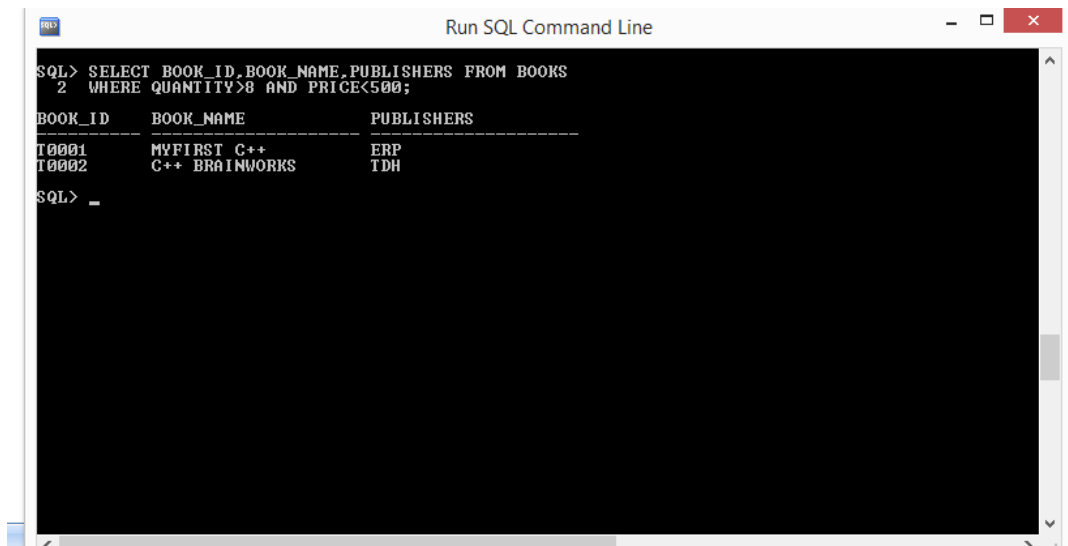


```
SQL> SELECT BOOK_NAME,AUTHOR_NAME,PRICE FROM BOOKS
2 WHERE PUBLISHERS=' FIRST PUBL';
```

BOOK_NAME	AUTHOR_NAME	PRICE
THE TEARS	WILLIAM HOPKINS	650
THUNDERBOLTS	ANA ROBERTS	750

```
SQL>
```

2. DISPLAY BOOK ID, BOOK NAME AND PUBLISHER OF BOOKS HAVING QUANTITY MORE THAN 8 AND PRICE LESS THAN 500

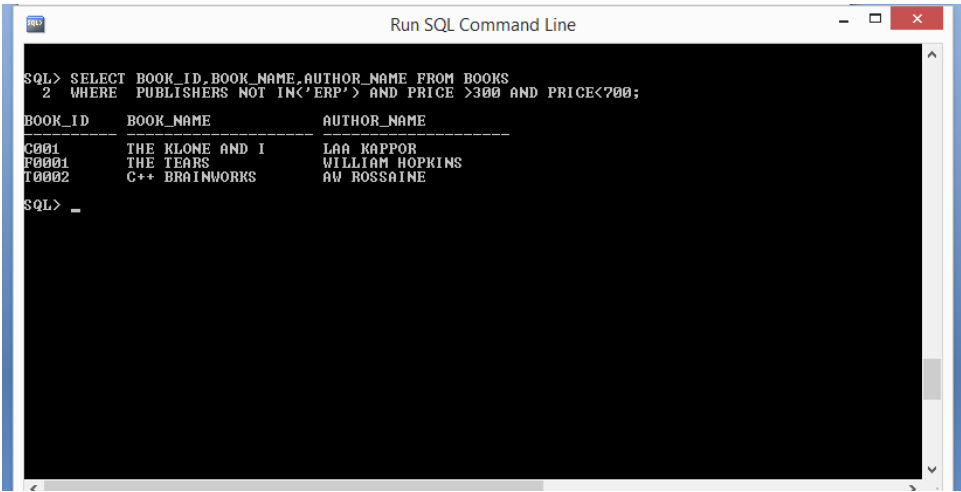


```
SQL> SELECT BOOK_ID,BOOK_NAME,PUBLISHERS FROM BOOKS
2 WHERE QUANTITY>8 AND PRICE<500;
```

BOOK_ID	BOOK_NAME	PUBLISHERS
T0001	MYFIRST C++	ERP
T0002	C++ BRAINWORKS	IDH

```
SQL> _
```

3. SELECT BOOK ID, BOOK NAME, AUTHOR NAME OF BOOKS WHICH IS PUBLISHED BY OTHER THAN ERP PUBLISHERS AND PRICE BETWEEN 300 TO 700.

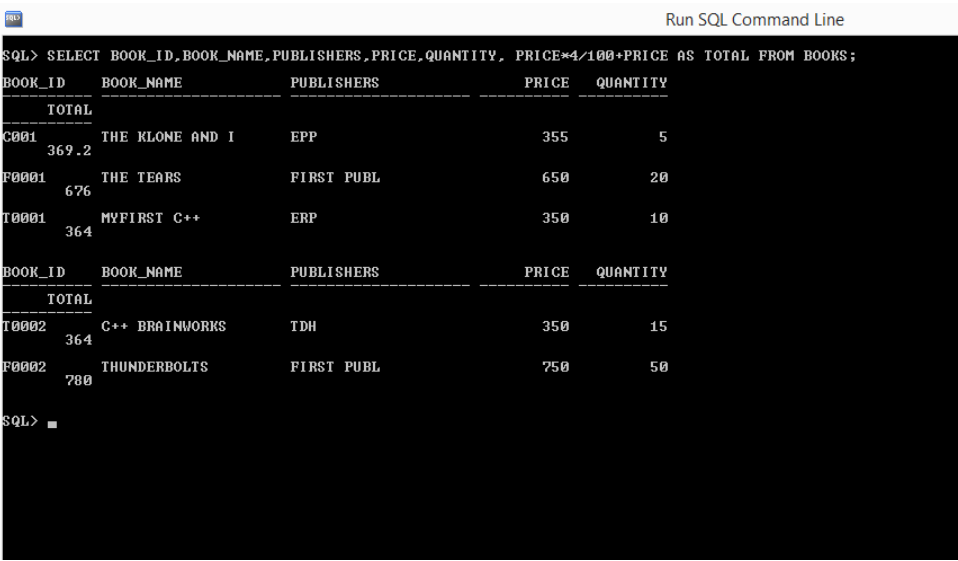


```
SQL> SELECT BOOK_ID,BOOK_NAME,AUTHOR_NAME FROM BOOKS
2 WHERE PUBLISHERS NOT IN('ERP') AND PRICE >300 AND PRICE<700;
```

BOOK_ID	BOOK_NAME	AUTHOR_NAME
C001	THE KLONE AND I	LAA KAPPOR
F0001	THE TEARS	WILLIAM HOPKINS
T0002	C++ BRAINWORKS	AV ROSSAINE

```
SQL> _
```

4. GENERATE A BILL WITH BOOK_ID, BOOK_NAME, PUBLISHER, PRICE, QUANTITY, 4% OF VAT —TOTAL||

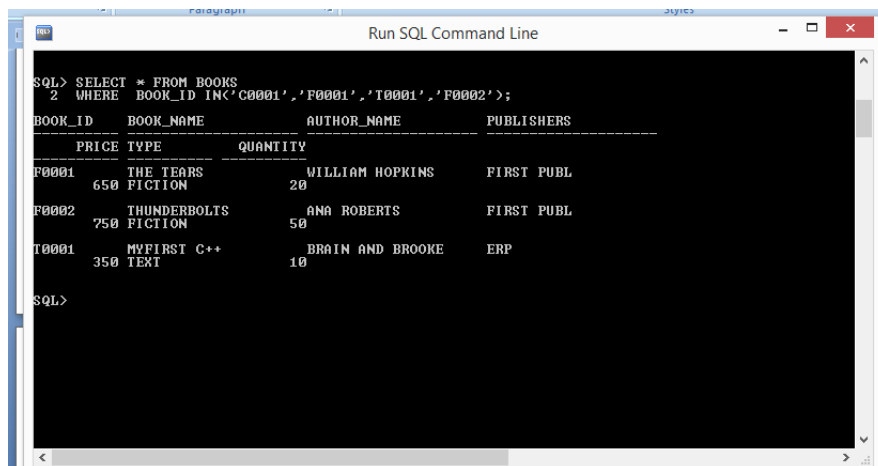


```
SQL> SELECT BOOK_ID,BOOK_NAME,PUBLISHERS,PRICE,QUANTITY, PRICE*4/100+PRICE AS TOTAL FROM BOOKS;
```

BOOK_ID	BOOK_NAME	PUBLISHERS	PRICE	QUANTITY
TOTAL				
C001	THE KLONE AND I	EPP	355	5
	369.2			
F0001	THE TEARS	FIRST PUBL	650	20
	676			
T0001	MYFIRST C++	ERP	350	10
	364			
TOTAL				
T0002	C++ BRAINWORKS	TDH	350	15
	364			
F0002	THUNDERBOLTS	FIRST PUBL	750	50
	780			

```
SQL> _
```

5. DISPLAY BOOK DETAILS WITH BOOK ID'S C0001, F0001, T0002, F0002
(HINT: USE IN OPERATOR)

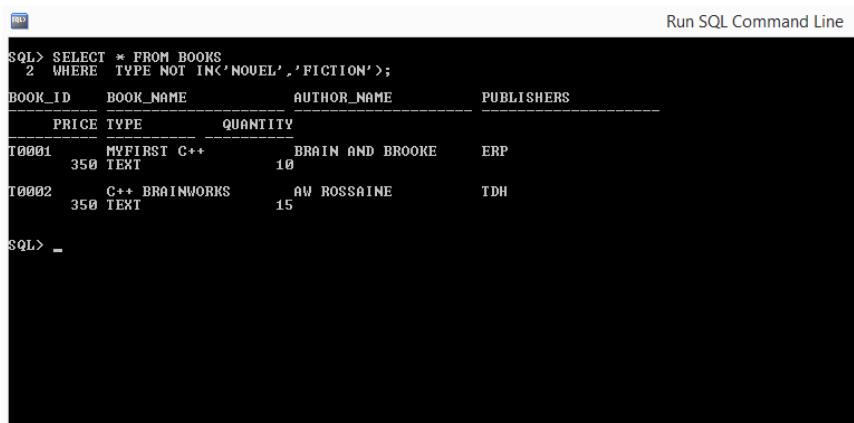


```
SQL> SELECT * FROM BOOKS
2 WHERE BOOK_ID IN('C0001','F0001','T0001','F0002');
```

BOOK_ID	BOOK_NAME	AUTHOR_NAME	PUBLISHERS
F0001	THE TEARS	WILLIAM HOPKINS	FIRST PUBL
F0002	THUNDERBOLTS	ANA ROBERTS	FIRST PUBL
T0001	MYFIRST C++	BRAIN AND BROOKE	ERP

```
SQL>
```

6. DISPLAY BOOK LIST OTHER THAN, TYPE NOVEL AND FICTION

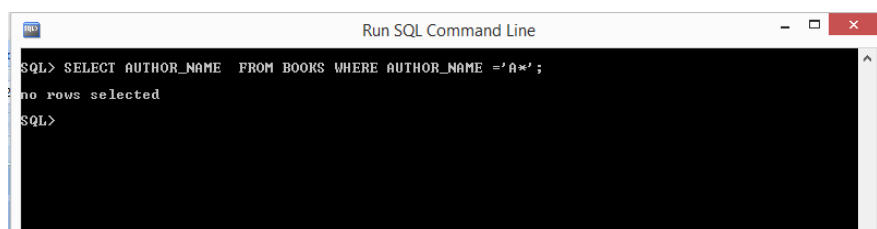


```
SQL> SELECT * FROM BOOKS
2 WHERE TYPE NOT IN('NOVEL','FICTION');
```

BOOK_ID	BOOK_NAME	AUTHOR_NAME	PUBLISHERS
T0001	MYFIRST C++	BRAIN AND BROOKE	ERP
T0002	C++ BRAINWORKS	AW ROSSAINE	IDH

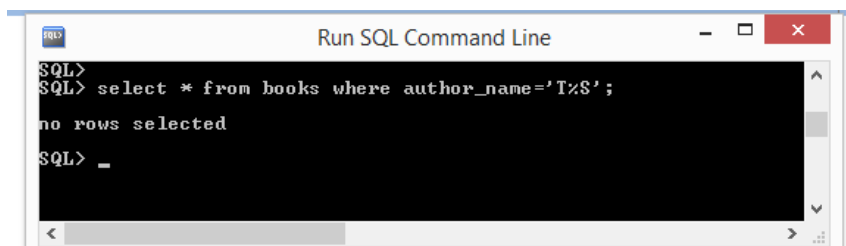
```
SQL> _
```

7. DISPLAY BOOK DETAILS WITH AUTHOR NAME STARTS WITH LETTER 'A'



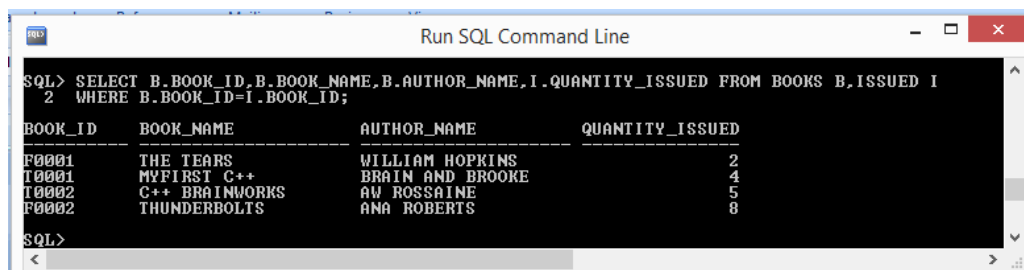
```
SQL> SELECT AUTHOR_NAME FROM BOOKS WHERE AUTHOR_NAME = 'A*';
no rows selected
SQL>
```

8. DISPLAY BOOK DETAILS WITH AUTHOR NAME STARTS WITH LETTER T AND ENDS WITH S



```
SQL>
SQL> select * from books where author_name='T%S';
no rows selected
SQL> _
```

9. SELECT BOOKID, BOOKNAME, AUTHOR NAME , QUANTITY ISSUED WHERE BOOKS.BOOKSID = ISSUED.BOOKID

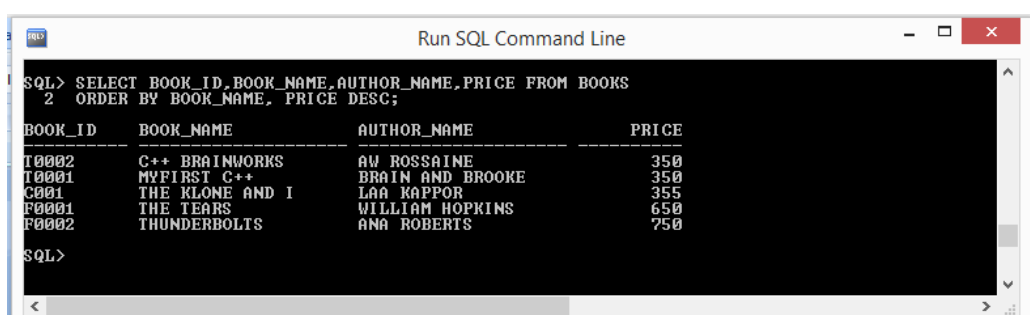


```
SQL> SELECT B.BOOK_ID,B.BOOK_NAME,B.AUTHOR_NAME,I.QUANTITY_ISSUED FROM BOOKS B,ISSUED I
2 WHERE B.BOOK_ID=I.BOOK_ID;
```

BOOK_ID	BOOK_NAME	AUTHOR_NAME	QUANTITY_ISSUED
P0001	THE TEARS	WILLIAM HOPKINS	2
T0001	MYFIRST C++	BRAIN AND BROOKE	4
T0002	C++ BRAINWORKS	AW ROSSAINE	5
P0002	THUNDERBOLTS	ANA ROBERTS	8

```
SQL>
```

10. LIST THE BOOK_NAME, AUTHOR_NAME, PRICE. IN ASCENDING ORDER OF BOOK_NAME AND THEN ON DESCENDING ORDER OF PRICE



```
SQL> SELECT BOOK_ID,BOOK_NAME,AUTHOR_NAME,PRICE FROM BOOKS
2 ORDER BY BOOK_NAME, PRICE DESC;
```

BOOK_ID	BOOK_NAME	AUTHOR_NAME	PRICE
T0002	C++ BRAINWORKS	AW ROSSAINE	350
T0001	MYFIRST C++	BRAIN AND BROOKE	350
C001	THE KLONE AND I	LAA KAPPOR	355
P0001	THE TEARS	WILLIAM HOPKINS	650
P0002	THUNDERBOLTS	ANA ROBERTS	750

```
SQL>
```

Activity 4: (Date Functions)**Database : Lab**

Create Following **table** and insert **tuples** with suitable constraints

Table : Equipment Details

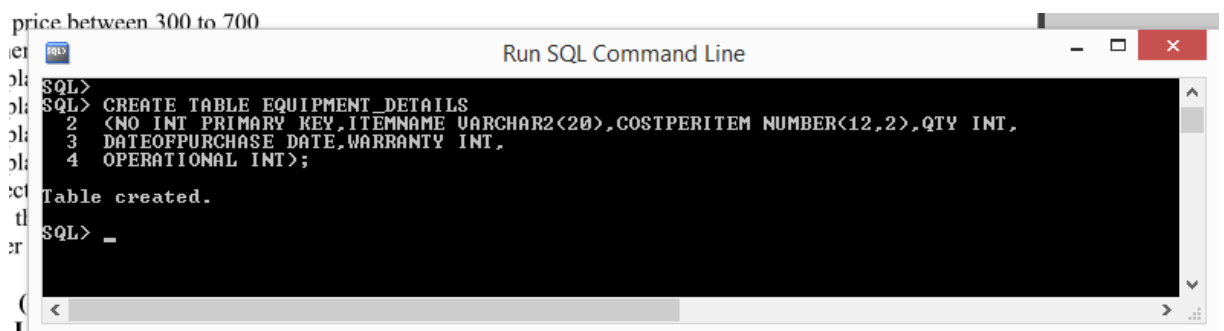
No	ItemName	Costperitem	Quantity	Dateofpurchase	Warranty	Operational
1	Computer	30000	9	21/5/07	2	7
2	Printer	5000	3	21/5/06	4	2
3	Scanner	8000	1	29/8/08	3	1
4	Camera	7000	2	13/6/05	1	2
5	UPS	15000	5	21/5/08	1	4
6	Hub	8000	1	31/10/08	2	1
7	Plotter	25000	2	11/1/09	2	2

(Use date functions and aggregate functions)

1. To select the ItemName purchase after 31/10/07
2. Extend the warrenty of each item by 6 months
3. Display Itemname , Dateof purchase and number of months between purchase date and present date
4. To list the ItemName in ascending order of the date of purchase where quantity is more than 3.
5. To count the number, average of costperitem of items purchased before 1/1/08
6. To display the minimum warranty , maximum warrenty period
7. To Display the day of the date , month , year of purchase in characters
8. To round of the warranty period to month and year format.
9. To display the next Sunday from the date '07-JUN-96'
10. To list the ItemNaName, which are within the warranty period till present date

TABLE CREATION – EQUIPMENT DETAILS

```
price between 300 to 700
```



The screenshot shows a 'Run SQL Command Line' window with a black background and white text. The SQL command to create the table is entered and executed successfully.

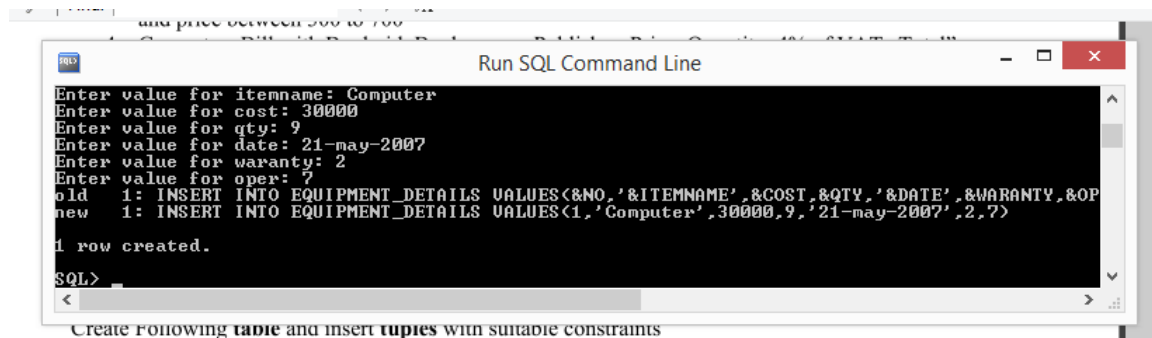
```
SQL> CREATE TABLE EQUIPMENT_DETAILS
2  (NO INT PRIMARY KEY, ITEMNAME VARCHAR2(20), COSTPERITEM NUMBER(12,2), QTY INT,
3   DATEOFPURCHASE DATE, WARRANTY INT,
4   OPERATIONAL INT);

Table created.

SQL> _
```

INSERTING RECORDS IN EQUIPMENT DETAILS

```
and price between 300 to 700
```



The screenshot shows a 'Run SQL Command Line' window with a black background and white text. It displays the prompts for inserting a record, the SQL INSERT statement, and the successful execution result.

```
Enter value for itemname: Computer
Enter value for cost: 30000
Enter value for qty: 9
Enter value for date: 21-may-2007
Enter value for warranty: 2
Enter value for oper: 7
old 1: INSERT INTO EQUIPMENT_DETAILS VALUES(&NO, '&ITEMNAME', &COST, &QTY, '&DATE', &WARRANTY, &OP
new 1: INSERT INTO EQUIPMENT_DETAILS VALUES(1, 'Computer', 30000, 9, '21-may-2007', 2, 7)

1 row created.

SQL> _
```

Create following **table** and insert **tuples** with suitable constraints

AFTER INSERTING ALL RECORDS IN EQUIPMENT DETAILS TABLE RECORDS IN TABLE ARE:

Run SQL Command Line

```
SQL> select * from equipment_details;
```

NO	ITEMNAME	COSTPERITEM	QTY	DATEOFFPUR	WARRANTY
1	Computer	30000	9	21-MAY-07	2
2	Printer	5000	3	21-MAY-06	4
3	Scanner	8000	1	29-AUG-08	3
4	Camera	7000	2	13-JUN-05	1
5	UPS	15000	5	21-MAY-08	1
6	Hub	8000	1	31-OCT-08	2
7	Plotter	25000	2	11-JAN-09	2

7 rows selected.
SQL> _

1. TO SELECT THE ITEMNAME PURCHASE AFTER 31/10/07

Run SQL Command Line

```
SQL> SELECT ITEMNAME FROM EQUIPMENT_DETAILS  
2 WHERE DATEOFFPURCHASE>'31-Aug-2007';
```

ITEMNAME
Scanner
UPS
Hub
Plotter

SQL> _

2. EXTEND THE WARRANTY OF EACH ITEM BY 6 MONTHS

Run SQL Command Line

```
SQL> UPDATE EQUIPMENT_DETAILS  
2 SET WARRANTY=WARRANTY+6;
```

7 rows updated.

```
SQL> select warranty from equipment_details;
```

WARRANTY
8
10
9
7
7
8
8

7 rows selected.
SQL> _

3. DISPLAY ITEMNAME , DATEOF PURCHASE AND NUMBER OF MONTHS BETWEEN PURCHASE DATE AND PRESENT DATE

```
Run SQL Command Line

SQL> SELECT ITEMNAME,DATEOFFPURCHASE,ROUND((SYSDATE-DATEOFFPURCHASE)/30,0) FROM EQUIPMENT_DETAILS;

ITEMNAME      DATEOFFPUR  ROUND((SYSDATE-DATEOFFPURCHASE)/30,0)
-----
Computer      21-MAY-07      144
Printer       21-MAY-06      156
Scanner       29-AUG-08      128
Camera        13-JUN-05      167
UPS           21-MAY-08      132
Hub           31-OCT-08      126
Plotter       11-JAN-09      124

7 rows selected.

SQL>
```

4. TO LIST THE ITEMNAME IN ASCENDING ORDER OF THE DATE OF PURCHASE WHERE QUANTITY IS MORE THAN 3.

```
Run SQL Command Line

SQL> SELECT ITEMNAME FROM EQUIPMENT_DETAILS
2 WHERE QTY>3 ORDER BY DATEOFFPURCHASE;

ITEMNAME
-----
Computer
UPS

SQL>
```

5. TO COUNT THE NUMBER, AVERAGE OF COSTPERITEM OF ITEMS PURCHASED BEFORE 1/1/08

```
Run SQL Command Line

ORA-00904: "COSTPERITEM": invalid identifier

SQL> SELECT COUNT(NO),AUG(COSTPERITEM) FROM EQUIPMENT_DETAILS WHERE DATEOFFPURCHASE<'01-JAN-2008';

COUNT(NO)  AUG(COSTPERITEM)
-----
4           14000

SQL>
```

6. TO DISPLAY THE MINIMUM WARRANTY, MAXIMUM WARRANTY PERIOD

```
Run SQL Command Line
ORA-00923: FROM keyword not found where expected

SQL> SELECT MAX(WARRANTY), MIN(WARRANTY) FROM EQUIPMENT_DETAILS;
MAX(WARRANTY) MIN(WARRANTY)
-----
10              7

SQL>
```

7. TO DISPLAY THE DAY OF THE DATE , MONTH , YEAR OF PURCHASE IN CHARACTERS.

```
Run SQL Command Line

SQL> SELECT TO_CHAR(DATEOFPURCHASE, 'DAY'), TO_CHAR(DATEOFPURCHASE, 'MON'), TO_CHAR(DATEOFPURCHASE, 'YEAR') FROM EQUIPMENT_DETAILS;
TO_CHAR(D TO_ TO_CHAR(DATEOFPURCHASE, 'YEAR')
-----
MONDAY    MAY TWO THOUSAND SEVEN
SUNDAY    MAY TWO THOUSAND SIX
FRIDAY    AUG TWO THOUSAND EIGHT
MONDAY    JUN TWO THOUSAND FIVE
WEDNESDAY MAY TWO THOUSAND EIGHT
FRIDAY    OCT TWO THOUSAND EIGHT
SUNDAY    JAN TWO THOUSAND NINE

7 rows selected.

SQL>
```

8. TO ROUND OF THE WARRANTY PERIOD TO MONTH AND YEAR FORMAT.

```
Run SQL Command Line

SQL> SELECT TO_CHAR(TO_DATE(WARRANTY, 'MM')) FROM EQUIPMENT_DETAILS;
TO_CHAR(T
-----
01-AUG-19
01-OCT-19
01-SEP-19
01-JUL-19
01-JUL-19
01-AUG-19
01-AUG-19

7 rows selected.

SQL>
```

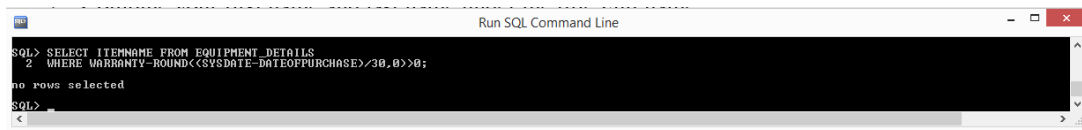
9. TO DISPLAY THE NEXT SUNDAY FROM THE DATE '07-JUN-96'

```
Run SQL Command Line

SQL> SELECT NEXT_DAY('7-JUN-1996', 'SUNDAY') FROM DUAL;
NEXT_DAY(
09-JUN-96

SQL>
```

**10. TO LIST THE ITEMNAME, WHICH ARE WITHIN THE WARRANTY PERIOD
TILL PRESENT DATE**

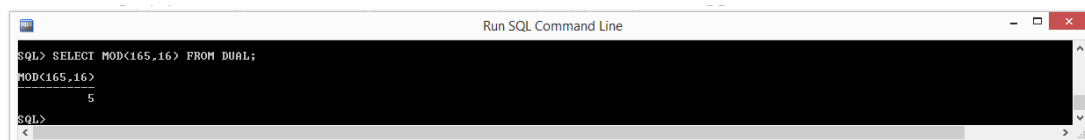


```
Run SQL Command Line
SQL> SELECT ITEMNAME FROM EQUIPMENT_DETAILS
2 WHERE WARRANTY-ROUND((SYSDATE-DATEOFPURCHASE)/30,0)>0;
no rows selected
SQL>
```

Activity 5: (Numeric, character functions)**Use Functions for the following**

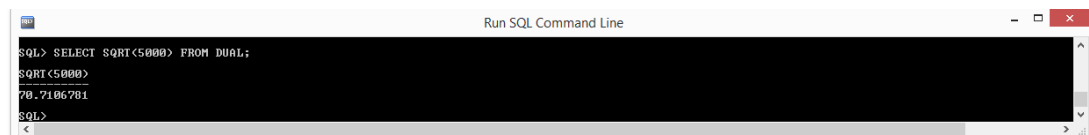
1. Find the mod of 165,16
2. Find Square Root of 5000
3. Truncate the value 128.3285 to 2 and -1 decimal places
4. Round the value 92.7683 to 2 and -1 decimal places
5. Convert the string '_Department' to uppercase and lowercase
6. Display your address convert the first character of each word to uppercase and rest are in lowercase
7. Combine your first name and last name under the title Full name
8. A) Take a string length maximum of 15 display your name to the left. The remaining space should be filled with '_'
9. Take a string length maximum of 20 display your name to the right. The remaining space should be filled with '_'
10. Find the length of the string '_JSS College, Mysore'
11. Display substring '_BASE' from '_DATABASE'
12. Display the position of the first occurrence of character '_' in Position and Length
13. Replace string Database with Datatype
14. Display the ASCII value of '_' (Space)
15. Display the Character equivalent of 42

1. FIND THE MOD OF 165,16



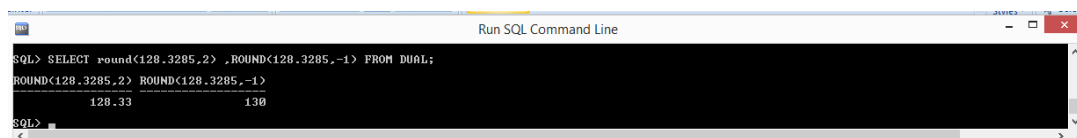
```
SQL> SELECT MOD(165,16) FROM DUAL;  
MOD(165,16)  
5  
SQL>
```

2. FIND SQUARE ROOT OF 5000



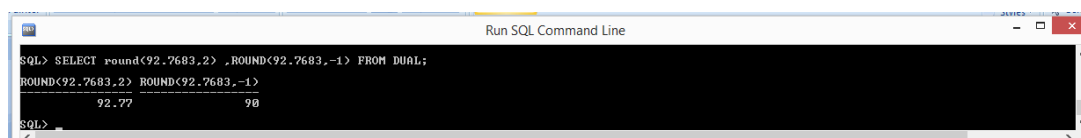
```
SQL> SELECT SQRT(5000) FROM DUAL;  
SQRT(5000)  
70.7106781  
SQL>
```

3. TRUNCATE THE VALUE 128.3285 TO 2 AND -1 DECIMAL PLACES



```
SQL> SELECT round(128.3285,2) ,ROUND(128.3285,-1) FROM DUAL;  
ROUND(128.3285,2) ROUND(128.3285,-1)  
128.33 130  
SQL>
```

4. ROUND THE VALUE 92.7683 TO 2 AND -1 DECIMAL PLACES



```
SQL> SELECT round(92.7683,2) ,ROUND(92.7683,-1) FROM DUAL;  
ROUND(92.7683,2) ROUND(92.7683,-1)  
92.77 90  
SQL>
```

5. CONVERT THE STRING 'DEPARTMENT' TO UPPERCASE AND LOWERCASE

```
Run SQL Command Line
SQL> SELECT UPPER('Department'), LOWER('Department') FROM DUAL;
UPPER<'DEPARTMENT'>
DEPARTMENT
SQL>
```

6. DISPLAY YOUR ADDRESS CONVERT THE FIRST CHARACTER OF EACH WORD TO UPPERCASE AND REST ARE IN LOWERCASE

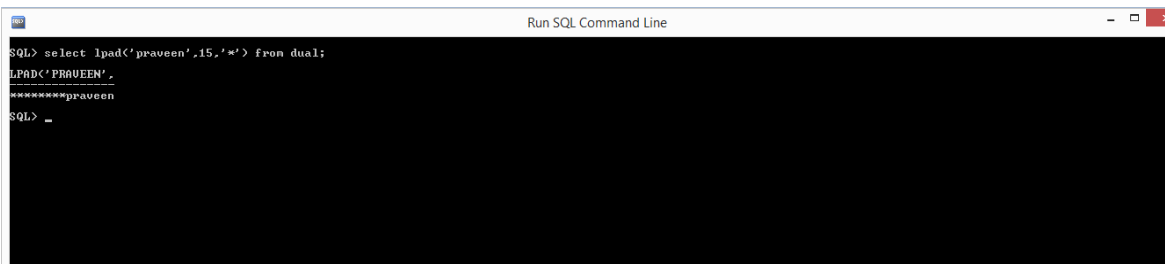
```
Run SQL Command Line
SQL*Plus: Release 10.2.0.1.0 - Production on Mon Mar 11 22:45:03 2019
Copyright (c) 1982, 2005, Oracle. All rights reserved.
SQL> connect scott/tiger
Connected.
SQL> select initcap('mg road') from dual;
INITCAP
Mg Road
SQL> SELECT INITCAP('mg road bangalore') FROM DUAL;
INITCAP('MGRoadBangalore')
Mg Road Bangalore
SQL>
```

7. COMBINE YOUR FIRST NAME AND LAST NAME UNDER THE TITLE FULL

```
Run SQL Command Line
SQL*Plus: Release 10.2.0.1.0 - Production on Mon Mar 11 22:45:03 2019
Copyright (c) 1982, 2005, Oracle. All rights reserved.
SQL> connect scott/tiger
Connected.
SQL> select initcap('mg road') from dual;
INITCAP
Mg Road
SQL> SELECT INITCAP('mg road bangalore') FROM DUAL;
INITCAP('MGRoadBangalore')
Mg Road Bangalore
SQL> SELECT CONCAT('JOHN', 'SIMPSON') FROM DUAL
CONCAT('JOHN', 'SIMPSON')
JOHNSIMPSON
SQL>
```

NAME

8. A) TAKE A STRING LENGTH MAXIMUM OF 15 DISPLAY YOUR NAME TO THE LEFT. THE REMAINING SPACE SHOULD BE FILLED WITH =



```
SQL> select lpad('praveen',15,'*') from dual;
LPAD('PRAVEEN',
*****praveen
SQL> _
```

9. TAKE A STRING LENGTH MAXIMUM OF 20 DISPLAY YOUR NAME TO THE RIGHT. THE REMAINING SPACE SHOULD BE FILLED WITH #



```
SQL> select lpad('praveen',15,'*') from dual;
LPAD('PRAVEEN',
*****praveen
SQL> select rpad('praveen',20,'#') from dual;
RPAD('PRAVEEN',20,'#'
praveen#####
SQL> _
```

10. FIND THE LENGTH OF THE STRING `JSS COLLEGE, MYSORE`

```
Run SQL Command Line
SQL> select lpad('praveen',15,'*') from dual;
LPAD('PRAVEEN',
*****praveen
SQL> select rpad('praveen',20,'#') from dual;
RPAD('PRAVEEN',20,'#
praveen#####
SQL> SELECT LENGTH('JSS College, Mysore') from dual;
LENGTH('JSSCOLLEGE,MYSORE')
19
SQL>
```

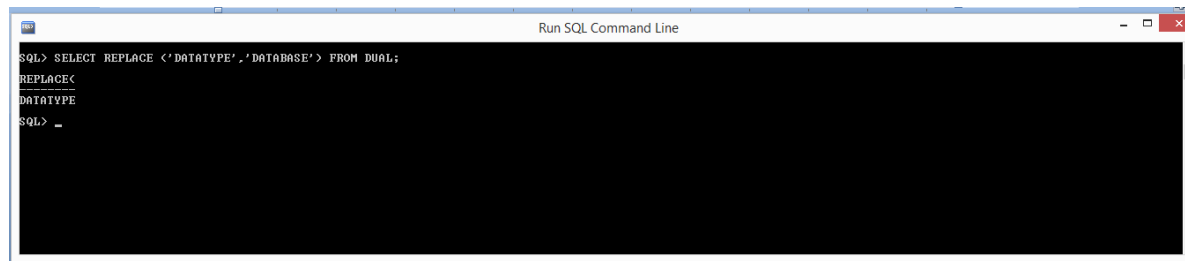
11. DISPLAY SUBSTRING `BASE` FROM `DATABASE`

```
Run SQL Command Line
SQL> select substr('Database',5,4) from dual;
SUBS
base
SQL>
```

12. DISPLAY THE POSITION OF THE FIRST OCCURRENCE OF CHARACTER `O` IN POSITION AND LENGTH

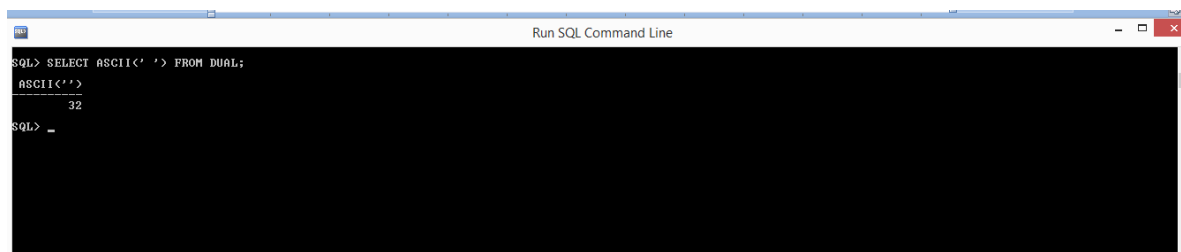
```
Run SQL Command Line
SQL> select instr('Operating system','o') from dual;
INSTR('OPERATINGSYSEM','O')
0
SQL>
```

13. REPLACE STRING DATABASE WITH DATATYPE



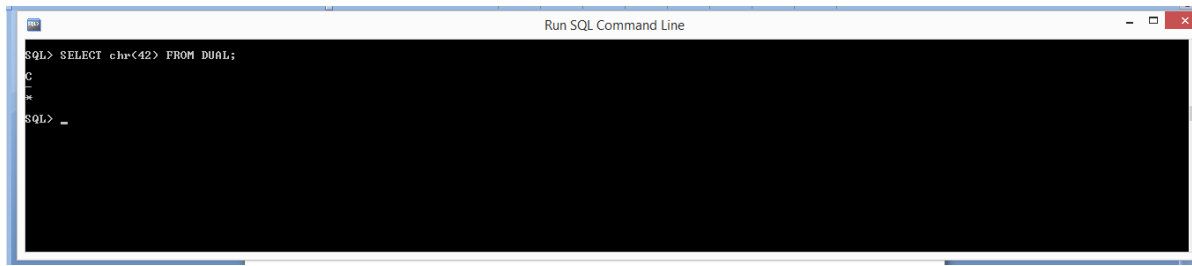
```
SQL> SELECT REPLACE ('DATATYPE','DATABASE') FROM DUAL;  
REPLACE(  
DATATYPE  
SQL> _
```

14. DISPLAY THE ASCII VALUE OF == (SPACE)



```
SQL> SELECT ASCII(' ') FROM DUAL;  
ASCII(' ')  
32  
SQL> _
```

15. DISPLAY THE CHARACTER EQUIVALENT OF 42



The screenshot shows a window titled "Run SQL Command Line". The command entered is "SQL> SELECT chr(42) FROM DUAL;". The output displayed is a single asterisk character (*).

```
SQL> SELECT chr(42) FROM DUAL;  
C  
*  
SQL> _
```

Activity : 6 (set operators)**Database : subject**

Create Following **table** and insert **tuples** with suitable constraints

Table - Physics

Regno	Name	Year	Combination
AJ00325	Ashwin	First	PCM
AJ00225	Swaroop	Second	PMCs
AJ00385	Sarika	Third	PME
AJ00388	Hamsa	First	PMCs

Table – Computer Science

Regno	Name	Year	Combination
AJ00225	Swaroop	Second	PMCs
AJ00296	Tajas	Second	BCA
AJ00112	Geetha	First	BCA
AJ00388	Hamsa	First	PMCs

1. Select all students from physics and Computer Science
2. Select student common in physics and Computer Science
3. Display all student details those are studying in second year
4. Display student those who are studying both physics and computer science in second year
5. Display the students studying only physics
6. Display the students studying only Computer Science
7. select all student having PMCs combination
8. select all student having BCA combination
9. select all student studying in Third year
10. Rename table Computer Science to CS

TABLE CREATION: PHYSICS TABLE

```
Run SQL Command Line

SQL> CREATE TABLE PHYSICS ( REGNO VARCHAR2(10) PRIMARY KEY,
  2 NAME VARCHAR2 (15), YEAR VARCHAR2(7),COMBINATION VARCHAR2(5));
Table created.

SQL> CREATE TABLE COMPUTERSCIENCE ( REGNO VARCHAR2(10), NAME VARCHAR2(15),
  2 YEAR VARCHAR2(7),COMBINATION VARCHAR2(5));
Table created.

SQL> _
```

TABLE CREATION: COMPUTERSCIENCE TABLE

```
Run SQL Command Line

SQL> CREATE TABLE PHYSICS ( REGNO VARCHAR2(10) PRIMARY KEY,
  2 NAME VARCHAR2 (15), YEAR VARCHAR2(7),COMBINATION VARCHAR2(5));
Table created.

SQL> CREATE TABLE COMPUTERSCIENCE ( REGNO VARCHAR2(10), NAME VARCHAR2(15),
  2 YEAR VARCHAR2(7),COMBINATION VARCHAR2(5));
Table created.

SQL> _
```

INSERTING RECORDS IN PHYSICS TABLE

```
Run SQL Command Line

SQL> SELECT P.NAME,C.NAME FROM PHYSICS P ,COMPUTERSCIENCE C
  2 WHERE P.REGNO=C.REGNO;
no rows selected

SQL> INSERT INTO PHYSICS VALUES('&REGNO','&NAME','&YEAR','&COMBI');
Enter value for regno: AJ00325
Enter value for name: Ashwin
Enter value for year: First
Enter value for combi: PCM
old 1: INSERT INTO PHYSICS VALUES('&REGNO','&NAME','&YEAR','&COMBI')
new 1: INSERT INTO PHYSICS VALUES('AJ00325','Ashwin','First','PCM')
1 row created.

SQL> /
Enter value for regno:
```

ALL RECORDS OF PHYSICS TABLE AFTER INSERTION

```
Run SQL Command Line

SQL> select * from Physics;

REGNO      NAME      YEAR      COMBI
-----
AJ00325    Ashwin    First     PCM
AJ00225    Svaroop   Second    PCM
AJ00385    Sarika    Third     PME
AJ00388    Hansa     First     PCM

SQL>
```

INSERTING RECORDS IN COMPUTER SCIENCE TABLE

```
Run SQL Command Line
SQL> INSERT INTO COMPUTERSCIENCE VALUES('&REGNO','&NAME','&YEAR','&COMBI');
Enter value for regno: AJ00225
Enter value for name: Swaroop
Enter value for year: Second
Enter value for combi: PMS
old 1: INSERT INTO COMPUTERSCIENCE VALUES('&REGNO','&NAME','&YEAR','&COMBI')
new 1: INSERT INTO COMPUTERSCIENCE VALUES('AJ00225','Swaroop','Second','PMS')
1 row created.
SQL> /
Enter value for regno: _
```

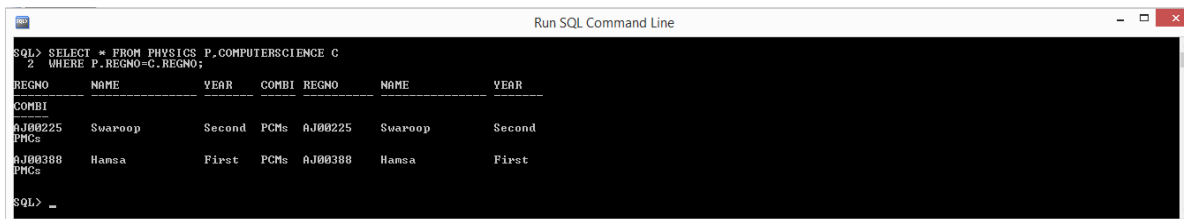
ALL RECORDS OF COMPUTERSCIENCE TABLE AFTER INSERTION

```
Run SQL Command Line
SQL> SELECT * FROM COMPUTERSCIENCE;
REGNO      NAME      YEAR      COMBI
-----
AJ00225    Swaroop    Second    PMS
AJ00296    Tajas     Second    BCR
AJ00112    Geetha     First     BCR
AJ00388    Hansa      First     PMS
SQL>
```

1. SELECT ALL STUDENTS FROM PHYSICS AND COMPUTER SCIENCE

```
Run SQL Command Line
SQL> SELECT * FROM PHYSICS P,COMPUTERSCIENCE C;
REGNO      NAME      YEAR      COMBI  REGNO      NAME      YEAR
-----
COMBI
AJ00325    Ashwin     First     PCM    AJ00225    Swaroop    Second
PMS
AJ00325    Ashwin     First     PCM    AJ00296    Tajas     Second
BCR
AJ00325    Ashwin     First     PCM    AJ00112    Geetha     First
BCR
REGNO      NAME      YEAR      COMBI  REGNO      NAME      YEAR
-----
COMBI
AJ00325    Ashwin     First     PCM    AJ00388    Hansa      First
PMS
AJ00225    Swaroop    Second    PMS    AJ00225    Swaroop    Second
PMS
AJ00225    Swaroop    Second    PMS    AJ00296    Tajas     Second
BCR
REGNO      NAME      YEAR      COMBI  REGNO      NAME      YEAR
-----
COMBI
AJ00225    Swaroop    Second    PMS    AJ00112    Geetha     First
BCR
```

2. SELECT STUDENT COMMON IN PHYSICS AND COMPUTER SCIENCE

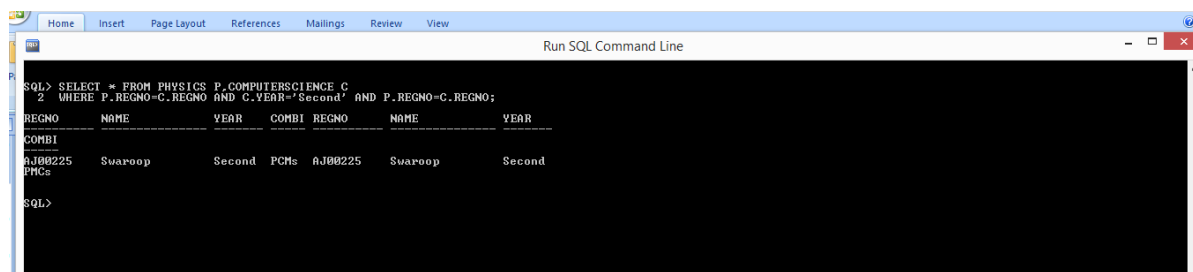


```
SQL> SELECT * FROM PHYSICS P,COMPUTERSCIENCE C
2 WHERE P.REGNO=C.REGNO;
```

REGNO	NAME	YEAR	COMBI	REGNO	NAME	YEAR
COMBI						
AJ00225	Swaroop	Second	PCMs	AJ00225	Swaroop	Second
AJ00388	Hansa	First	PCMs	AJ00388	Hansa	First

SQL>

3. DISPLAY ALL STUDENT DETAILS THOSE ARE STUDYING IN SECOND YEAR

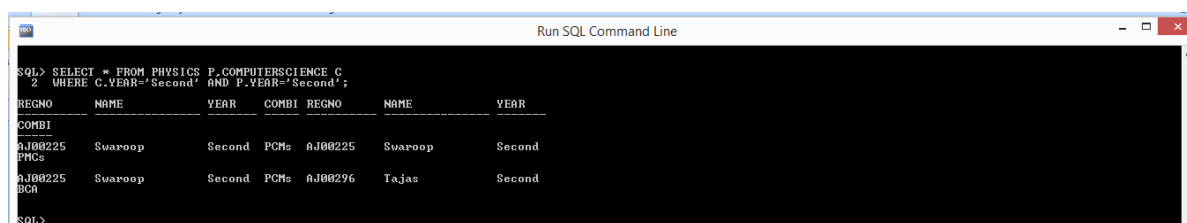


```
SQL> SELECT * FROM PHYSICS P,COMPUTERSCIENCE C
2 WHERE P.REGNO=C.REGNO AND C.YEAR='Second' AND P.REGNO=C.REGNO;
```

REGNO	NAME	YEAR	COMBI	REGNO	NAME	YEAR
COMBI						
AJ00225	Swaroop	Second	PCMs	AJ00225	Swaroop	Second

SQL>

4. DISPLAY STUDENT THOSE WHO ARE STUDYING BOTH PHYSICS AND COMPUTER SCIENCE IN SECOND YEAR

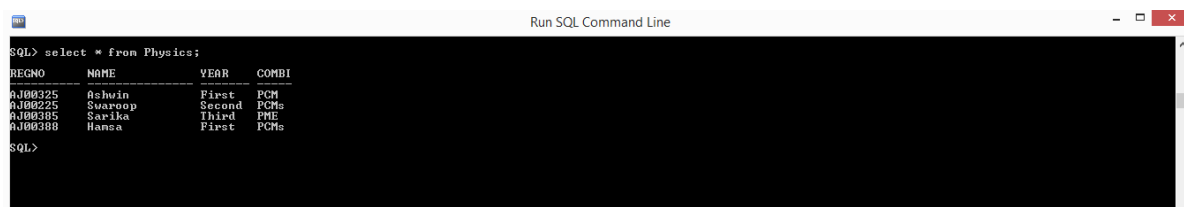


```
SQL> SELECT * FROM PHYSICS P,COMPUTERSCIENCE C
2 WHERE C.YEAR='Second' AND P.YEAR='Second';
```

REGNO	NAME	YEAR	COMBI	REGNO	NAME	YEAR
COMBI						
AJ00225	Swaroop	Second	PCMs	AJ00225	Swaroop	Second
AJ00225	Swaroop	Second	PCMs	AJ00296	Tajas	Second

SQL>

5. DISPLAY THE STUDENTS STUDYING ONLY PHYSICS



```
SQL> select * from Physics;
```

REGNO	NAME	YEAR	COMBI
AJ00325	Ashwin	First	PCM
AJ00225	Swaroop	Second	PCMs
AJ00388	Sarika	Third	PME
AJ00388	Hansa	First	PCMs

SQL>

6. DISPLAY THE STUDENTS STUDYING ONLY COMPUTER SCIENCE

```
SQL> SELECT * FROM COMPUTERSCIENCE;
REGNO      NAME      YEAR      COMBI
-----
AJ00225    Svaroop    Second    PMCs
AJ00296    Tajas      Second    BCA
AJ00112    Geetha     First     BCA
AJ00308    Hansa      First     PMCs
SQL>
```

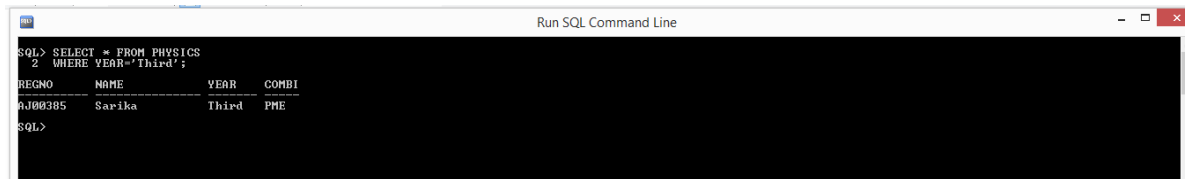
7. SELECT ALL STUDENT HAVING PMCS COMBINATION

```
SQL> SELECT * FROM PHYSICS P,COMPUTERSCIENCE C
2 WHERE P.REGNO=C.REGNO;
REGNO      NAME      YEAR      COMBI  REGNO      NAME      YEAR
-----
COMBI
AJ00225    Svaroop    Second    PMCs  AJ00225    Svaroop    Second
AJ00308    Hansa      First     PMCs  AJ00308    Hansa      First
SQL>
SQL>
SQL>
SQL>
```

8. SELECT ALL STUDENT HAVING BCA COMBINATION

```
SQL> SELECT * FROM COMPUTERSCIENCE
2 WHERE COMBINATION='BCA';
REGNO      NAME      YEAR      COMBI
-----
AJ00296    Tajas      Second    BCA
AJ00112    Geetha     First     BCA
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

9. SELECT ALL STUDENT STUDYING IN THIRD YEAR



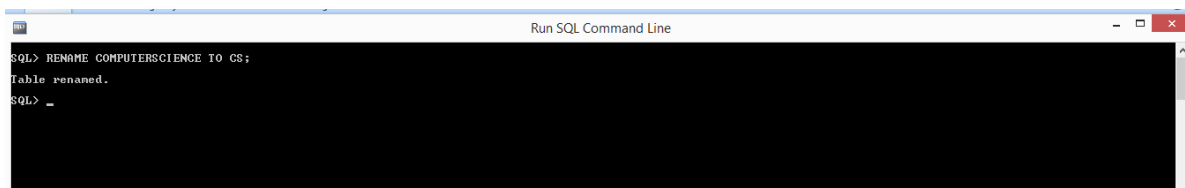
A screenshot of a 'Run SQL Command Line' window. The command entered is 'SQL> SELECT * FROM PHYSICS 2 WHERE YEAR='Third';'. The output shows a table with four columns: REGNO, NAME, YEAR, and COMBI. The first row of data has the values 'AJ00385', 'Sarika', 'Third', and 'PME'.

```
SQL> SELECT * FROM PHYSICS
2 WHERE YEAR='Third';
```

REGNO	NAME	YEAR	COMBI
AJ00385	Sarika	Third	PME

```
SQL>
```

10. RENAME TABLE COMPUTER SCIENCE TO CS



A screenshot of a 'Run SQL Command Line' window. The command entered is 'SQL> RENAME COMPUTERSCIENCE TO CS;'. The output shows 'Table renamed.' followed by a prompt character '_'.

```
SQL> RENAME COMPUTERSCIENCE TO CS;
```

```
Table renamed.
```

```
SQL> _
```


Activity 7: (views)**Database: Railway Reservation System**

Create Following **table** and insert **tuples** with suitable constraints

Table: Train Details

Train_no	Train_name	Start_place	Destination
RJD16	Rajdhani Express	Bangalore	Mumbai
UDE04	Udhyan Express	Chennai	Hyderabad
KKE55	Karnataka Express	Bangalore	Chennai
CSE3	Shivaji Express	Coimbatore	Bangalore
JNS8	Janashatabdi	Bangalore	Salem

Table : Availability

Train_no	Class	Start_Place	Destination	No_of_seats
RJD16	Sleeper Class	Banglore	Mumbai	15
UDE04	First Class	Chennai	Hyderabad	22
KKE55	First Class AC	Bangalore	Chennai	15
CSE3	Second Class	Coimbatore	Bangalore	8
JNS8	Sleeper Class	Bangalore	Salem	18

1. Create view **sleeper** to display train no, start place, destination which have sleeper class and perform the following
 - a. insert new record
 - b. update destination='Manglore' where train no='RJD16'
 - c. delete a record which have train no='KKE55'
2. Create view **details** to display train no, train name, class
3. Create view **total_seats** to display train number, start place, use count function to no of seats , group by start place and perform the following
 - a. insert new record
 - b. update start place='Hubli' where train no='JNS8'
 - c. delete last row of the view
4. Rename view sleeper to class
5. Delete view details

TABLE CREATION: TRAINDETAILS

```
Run SQL Command Line

SQL> CREATE TABLE TRAINDETAILS(
  2 TRAIN_NO VARCHAR2(7) PRIMARY KEY,
  3 TRAIN_NAME VARCHAR2(15),
  4 START_PLACE VARCHAR2(15),
  5 DESTINATION VARCHAR2(15));
Table created.
SQL> _
```

TABLE CREATION: AVAILABILITY

```
Run SQL Command Line

SQL> CREATE TABLE AVAILABILITY(
  3 CLASS1 VARCHAR2(15),
  4 START_PLACE VARCHAR2(12),
  5 DESTINATION VARCHAR2(15), NO_OF_SEATS INT);
Table created.
SQL> _
```

RECORD INSERTION : TRAINDETAILS TABLE

```
Run SQL Command Line

1 row created.
SQL> /
Enter value for trainno: UDEB4
Enter value for trainname: Udhyan Express
Enter value for startplace: Chennai
Enter value for destination: Hyderabad
old 1: INSERT INTO TRAINDETAILS VALUES('&TRAINNO','&TRAINNAME','&STARTPLACE','&DESTI')
new 1: INSERT INTO TRAINDETAILS VALUES('UDEB4','Udhyan Express','Chennai','Hyderabad')
1 row created.
SQL> _
```

AFTER INSERTING ALL RECORDS IN TRAINDETAILS TABLE

```
Run SQL Command Line

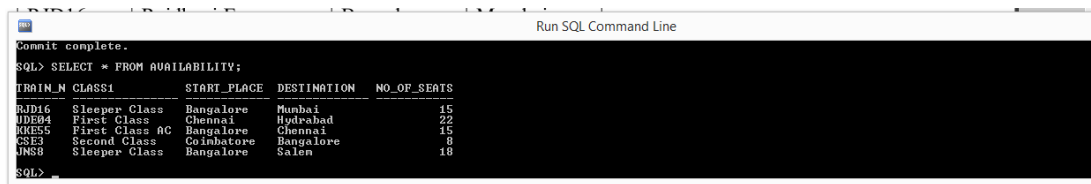
Commit complete.
SQL> SELECT * FROM TRAINDETAILS;
TRAIN_N TRAIN_NAME      START_PLACE  DESTINATION
-----
RJD16   Rajadhani Express   Bangalore    Mumbai
UDEB4   Udhyan Express      Chennai      Hyderabad
MKE55   Maranataka Express  Bangalore    Chennai
CSE3    Shivaaji Express    Coimbatore   Bangalore
UN88    Janashatabdi        Bangalore    Salem
SQL> _
```

INSERTING RECORDS IN AVAILABILITY TABLE

```
Run SQL Command Line

SQL> INSERT INTO AVAILABILITY VALUES (&TRAINNO,&CLASS,&STARTPLACE,&DESTI,&NOSEATS);
Enter value for trainno: RJD16
Enter value for class: Sleeper Class
Enter value for startplace: Bangalore
Enter value for destination: Mumbai
Enter value for noofseats: 15
old 1: INSERT INTO AVAILABILITY VALUES (&TRAINNO,&CLASS,&STARTPLACE,&DESTI,&NOSEATS)
new 1: INSERT INTO AVAILABILITY VALUES ('RJD16','Sleeper Class','Bangalore','Mumbai',15)
1 row created.
SQL> /
Enter value for trainno:
```

AFTER INSERTING ALL RECORDS IN AVAILABILITY TABLE



```
Commit complete.
SQL> SELECT * FROM AVAILABILITY;

```

TRAIN_N	CLASS1	START_PLACE	DESTINATION	NO_OF_SEATS
RJD16	Sleeper Class	Bangalore	Mumbai	15
UDE94	First Class	Chennai	Hydrabad	22
KKE55	First Class AC	Bangalore	Chennai	15
CSE1	Second Class	Coimbatore	Bangalore	8
JNS8	Sleeper Class	Bangalore	Salem	18

```
SQL>
```

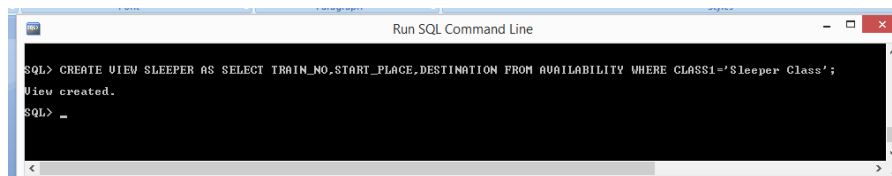
1. CREATE VIEW SLEEPER TO DISPLAY TRAIN NO, START PLACE, DESTINATION WHICH HAVE SLEEPER CLASS AND PERFORM THE FOLLOWING

A. INSERT NEW RECORD

B. UPDATE DESTINATION='MANGLORE' WHERE TRAIN NO='RJD16'

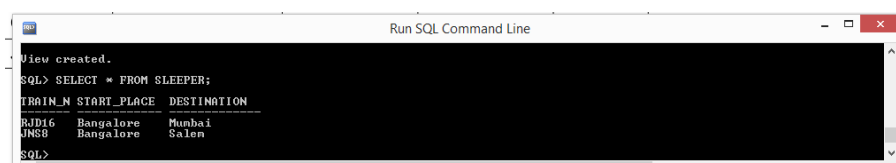
C. DELETE A RECORD WHICH HAVE TRAIN NO='KKE55'

CREATING SLEEPER VIEW



```
SQL> CREATE VIEW SLEEPER AS SELECT TRAIN_NO,START_PLACE,DESTINATION FROM AVAILABILITY WHERE CLASS1='Sleeper Class';
View created.
SQL>
```

CONTENT OF SLEEPER VIEW



```
View created.
SQL> SELECT * FROM SLEEPER;

```

TRAIN_N	START_PLACE	DESTINATION
RJD16	Bangalore	Mumbai
JNS8	Bangalore	Salem

```
SQL>
```

A. INSERT NEW RECORD

```
Run SQL Command Line
SQL> INSERT INTO SLEEPER VALUES('CSE3', 'Chennai','Bangalore');
1 row created.
SQL>
```

B. UPDATE DESTINATION='MANGLORE' WHERE TRAIN NO='RJD16'

```
Run SQL Command Line
SQL> SELECT * FROM SLEEPER;
TRAIN_N START_PLACE DESTINATION
-----
RJD16    Bangalore    Mumbai
JNSB     Bangalore    Salem

SQL> UPDATE SLEEPER SET DESTINATION='Mangalore' WHERE TRAIN_NO='RJD16';
1 row updated.
SQL>
```

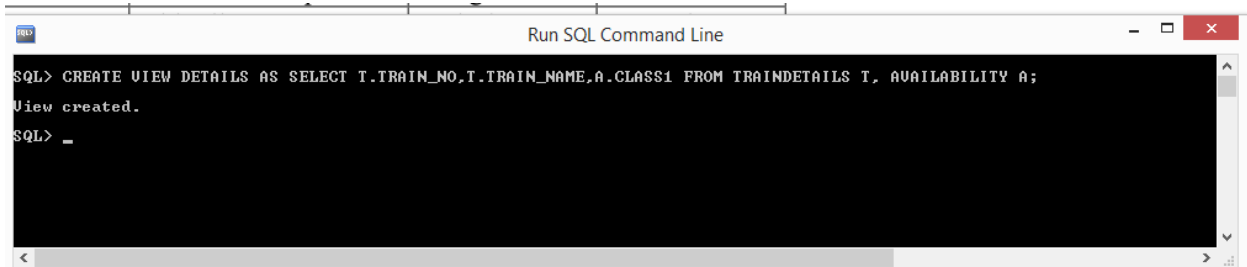
C. DELETE A RECORD WHICH HAVE TRAIN NO='KKE55'

```
Run SQL Command Line
JDEB4    First Class    Chennai    Hyderabad    22
KKE55    First Class AC    Bangalore    Chennai      15
CSE3     Second Class    Coimbatore    Bangalore     8
JNSB     Sleeper Class    Bangalore    Salem       18
CSE3     Sleeper Class    Chennai     Bangalore

6 rows selected.

SQL> DELETE FROM AVAILABILITY WHERE TRAIN_NO='KKE55';
1 row deleted.
SQL>
```

2. CREATE VIEW DETAILS TO DISPLAY TRAIN NO, TRAIN NAME, CLASS



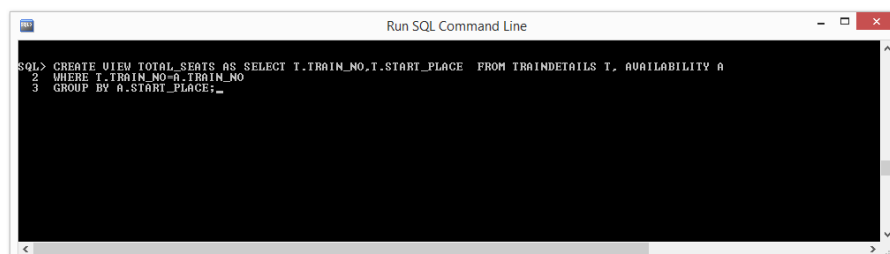
```
SQL> CREATE VIEW DETAILS AS SELECT T.TRAIN_NO,T.TRAIN_NAME,A.CLASS1 FROM TRAINDETAILS T, AVAILABILITY A;
View created.
SQL> _
```

3. CREATE VIEW TOTAL_SEATS TO DISPLAY TRAIN NUMBER, START PLACE, USE COUNT FUNCTION TO NO OF SEATS , GROUP BY START PLACE AND PERFORM THE FOLLOWING

A. INSERT NEW RECORD

B. UPDATE START PLACE='HUBLI' WHERE TRAIN NO='JNS8'

C. DELETE LAST ROW OF THE VIEW



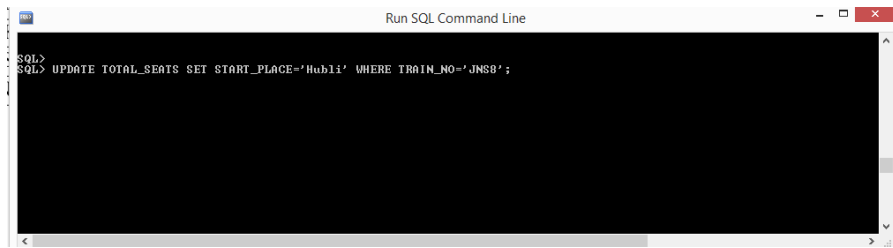
```
SQL> CREATE VIEW TOTAL_SEATS AS SELECT T.TRAIN_NO,T.START_PLACE FROM TRAINDETAILS T, AVAILABILITY A
2 WHERE T.TRAIN_NO=A.TRAIN_NO
3 GROUP BY A.START_PLACE;_
```

A. INSERT NEW RECORD



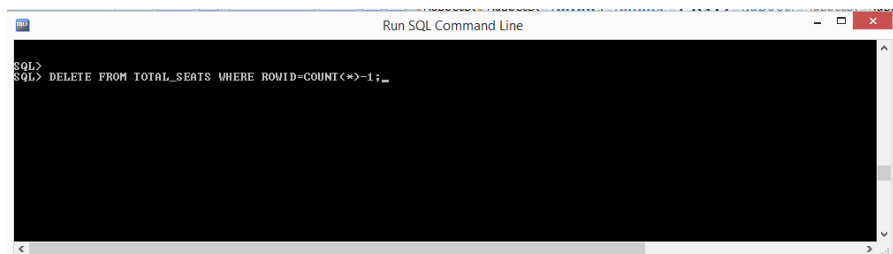
```
SQL>
SQL> INSERT INTO TOTAL_SEATS VALUES('CSE3','Shivaji Express','4');_
```

B. UPDATE START PLACE='HUBLI' WHERE TRAIN NO='JNS8'



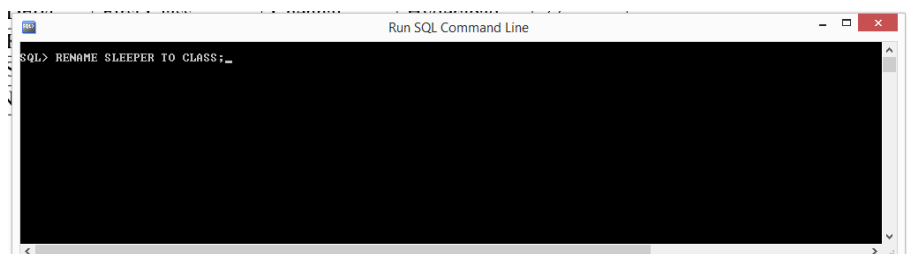
```
SQL>
SQL> UPDATE TOTAL_SEATS SET START_PLACE='Hubli' WHERE TRAIN_NO='JNS8';
```

C. DELETE LAST ROW OF THE VIEW



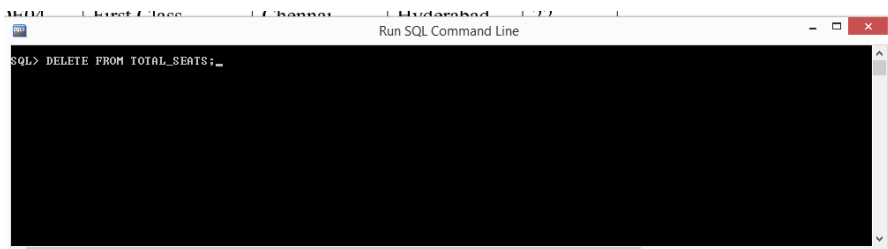
```
SQL>
SQL> DELETE FROM TOTAL_SEATS WHERE ROWID=COUNT(*)-1;_
```

4. RENAME VIEW SLEEPER TO CLASS



```
SQL> RENAME SLEEPER TO CLASS;_
```

5. DELETE VIEW DETAILS



```
SQL> DELETE FROM TOTAL_SEATS;_
```

DELETING SLEEPER TABLE

NOTE: SINCE TABLE NAME HAS BEEN CHANGED TO CLASS SO TO DELETE SLEEPER TABLE WE HAVE TO DELETE CLASS TABLE.

