

SISTER NIVEDITA UNIVERSITY



DATABASE MANAGEMENT SYSTEM

SUBMITTED BY: PIYUSH CHANDRA CHANDRA

DEPT- BTECH (CSE), ROLL- 1027

SUBMITTED TO: SWARUP KUMAR GHOSH &

DEBANJAN DAS

INDEX

S.NO	DESCRIPTION	PAGE NO	REMARKS
1	Assignment 5	4-14	
2	Assignment 6	16-47	
3	Assignment 7	49-62	

ASSIGNMENT 5

10/11/2020 - 17/11/2020

1. List the various CATEGORIES and COUNT OF BOOKS in each category.
- SELECT CATEGORY, COUNT(BOOK_NO) FROM BOOK GROUP BY CATEGORY;

```

SELECT CATEGORY, COUNT(BOOK_NO) FROM BOOK GROUP BY CATEGORY
SELECT MEMBER_ID,COUNT(BOOK_NO)FROM LIB_ISSUE_ID_500917 GROUP BY MEMBER_ID ORDER BY COUNT(*) DESC;
SELECT MEMBER_ID,COUNT(BOOK_NO)FROM LIB_ISSUE_ID_500917 GROUP BY MEMBER_ID HAVING COUNT(BOOK_NO)>2 ORDER BY COUNT(BOOK_NO);
SELECT MEMBER_ID,BOOK_NO,COUNT(BOOK_NO)FROM LIB_ISSUE_ID_500917 GROUP BY MEMBER_ID,BOOK_NO ORDER BY COUNT(BOOK_NO) DESC;
CONNECT /@ORCL/cems@12c$cse$;
CREATE USER user1 IDENTIFIED BY "idms@12";
GRANT CONNECT TO user1;
GRANT EXECUTE ON schema.procedure TO user1;
  
```

CATEGORY	COUNT(BOOK_NO)
Others	3
System	1
Science	1
Database	3

4 rows returned in 0.16 seconds Download

2. List the BOOK_NO and the NUMBER OF TIMES the Book is issued in the descending order of COUNT.

- SELECT Book_No, COUNT(Book_No) AS Number_Of_Time_Issued
FROM ISSUE GROUP BY Book_No ORDER BY Number_Of_Time_Issued DESC

```

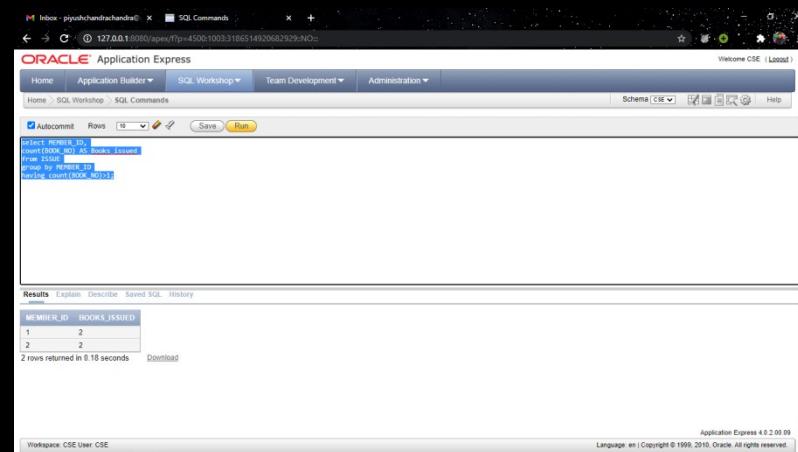
SELECT CATEGORY,COUNT(BOOK_NO) FROM BOOK GROUP BY CATEGORY;
SELECT Book_No, COUNT(Book_No) AS Number_Of_Time_Issued FROM ISSUE GROUP BY Book_No ORDER BY Number_Of_Time_Issued DESC;
  
```

BOOK_NO	NUMBER_OF_TIME_ISSUED
101	3
104	2
106	1
102	1

4 rows returned in 0.05 seconds Download

3. Display the MEMBER ID and the NO OF BOOKS for each member that has issued more than 1 books.

➤ select MEMBER_ID,
 count(BOOK_NO) AS Books_issued
 from ISSUE
 group by MEMBER_ID
 having count(BOOK_NO)>1;



The screenshot shows the Oracle Application Express interface with a SQL command window. The SQL code is:

```
select MEMBER_ID,
       count(BOOK_NO) AS Books_issued
  from ISSUE
 group by MEMBER_ID
 having count(BOOK_NO)>1;
```

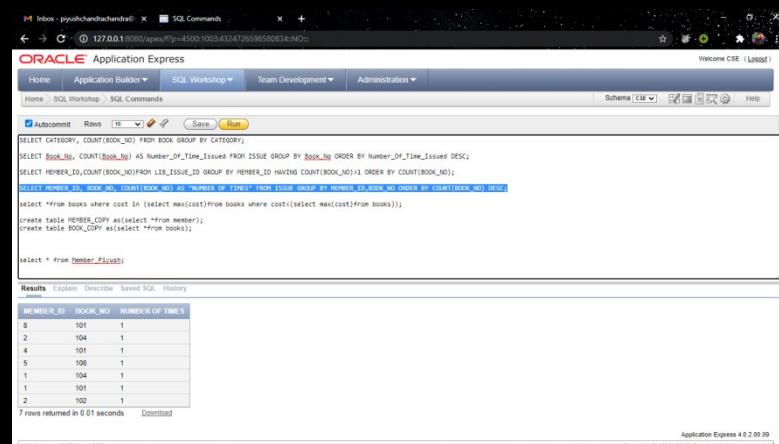
The results table shows:

MEMBER_ID	BOOKS_ISSUED
1	2
2	2

2 rows returned in 0.18 seconds

4. Display the MEMBER ID, BOOK NO and NO OF TIMES the same book is issued by the MEMBER in the descending order of count.

➤ SELECT MEMBER_ID, BOOK_NO, COUNT(BOOK_NO) AS "NUMBER OF TIMES" FROM ISSUE GROUP BY MEMBER_ID,BOOK_NO ORDER BY COUNT(BOOK_NO) DESC;



The screenshot shows the Oracle Application Express interface with a SQL command window containing a complex multi-step query:

```

SELECT CATEGORY, COUNT(BOOK_NO) FROM BOOK GROUP BY CATEGORY;
SELECT Book_No, COUNT(Book_No) AS Number_of_Time_Issued FROM ISSUE GROUP BY Book_No ORDER BY Number_of_Time_Issued DESC;
SELECT MEMBER_ID,COUNT(BOOK_NO)FROM ISSUE GROUP BY MEMBER_ID HAVING COUNT(BOOK_NO)>1 ORDER BY COUNT(BOOK_NO);
SELECT MEMBER_ID,BOOK_NO,COUNT(BOOK_NO) AS "NUMBER OF TIMES" FROM ISSUE GROUP BY MEMBER_ID,BOOK_NO ORDER BY COUNT(BOOK_NO) DESC;
select *from books where cost in (select max(cost)from books where cost=(select max(cost)from books));
create table MEMBER_COPY as(select *from member);
create table BOOK_COPY as(select *from books);

select * from Member_Pivot;
  
```

The results table shows:

MEMBER_ID	BOOK_NO	NUMBER OF TIMES
8	104	1
2	104	1
4	101	1
5	108	1
1	104	1
1	101	1
2	102	1

7 rows returned in 0.01 seconds

5. Select all information of Book with 2nd Maximum Cost.

- SELECT * FROM BOOK WHERE COST IN (SELECT MAX(COST) FROM BOOK WHERE COST < (SELECT MAX(COST) FROM BOOK));

The screenshot shows the Oracle Application Express interface with the following SQL code:

```

SELECT CATEGORY, COUNT(BOOK_NO) FROM BOOK GROUP BY CATEGORY;
SELECT Book_No, COUNT(BOOK_No) AS Number_of_Time_Issued FROM ISSUE GROUP BY Book_No ORDER BY Number_of_Time_Issued DESC;
SELECT MEMBER_ID,COUNT(BOOK_No)FROM LIB_ISSUE_ID GROUP BY MEMBER_ID HAVING COUNT(BOOK_No)>1 ORDER BY COUNT(BOOK_No);
SELECT MEMBER_ID, BOOK_No, COUNT(BOOK_No) AS "NUMBER OF TIMES" FROM ISSUE GROUP BY MEMBER_ID,BOOK_No ORDER BY COUNT(BOOK_No) DESC;
SELECT * FROM BOOK WHERE COST IN (SELECT MAX(COST) FROM BOOK WHERE COST < (SELECT MAX(COST) FROM BOOK));
Create table MEMBER_COPY as(select *from member);
Create table BOOK_COPY as(select *from books);

select * from member_Piyush;

```

The results table shows one row:

BOOK_NO	BOOK_NAME	AUTHOR_NAME	COST	CATEGORY
103	Visual Basic 10	EPB	700	Others

6. Copy the MEMBER, BOOKS and ISSUE tables and MEMBER_COPY, BOOKS_COPY, ISSUE_COPY respectively.

- CREATE TABLE MEMBER_COPY AS(SELECT * FROM Member_Piyush);
CREATE TABLE BOOK_COPY AS(SELECT * FROM BOOK);
CREATE TABLE MEMBER_COPY1 AS(SELECT * FROM Member_Piyush WHERE MEMBER_ADDRESS = 'Kolkata');
SELECT * FROM MEMBER_COPY1;

The screenshot shows the Oracle Application Express interface with the following SQL code:

```

SELECT CATEGORY, COUNT(BOOK_No) FROM BOOK GROUP BY CATEGORY;
SELECT Book_No, COUNT(BOOK_No) AS Number_of_Time_Issued FROM ISSUE GROUP BY Book_No ORDER BY Number_of_Time_Issued DESC;
SELECT MEMBER_ID,COUNT(BOOK_No)FROM LIB_ISSUE_ID GROUP BY MEMBER_ID HAVING COUNT(BOOK_No)>1 ORDER BY COUNT(BOOK_No);
SELECT MEMBER_ID, BOOK_No, COUNT(BOOK_No) AS "NUMBER OF TIMES" FROM ISSUE GROUP BY MEMBER_ID,BOOK_No ORDER BY COUNT(BOOK_No) DESC;
SELECT * FROM BOOK WHERE COST IN (SELECT MAX(COST) FROM BOOK WHERE COST < (SELECT MAX(COST) FROM BOOK));
Create table MEMBER_COPY as(select *from member);
Create table BOOK_COPY as(select *from books);

CREATE TABLE MEMBER_COPY1 AS (SELECT * FROM Member_Piyush WHERE MEMBER_ADDRESS = 'Kolkata');
SELECT * FROM MEMBER_COPY1;

Create table QUERY1 as select BOOK_No, ISSUE_DATE From issue;
Select * from QUERY1;

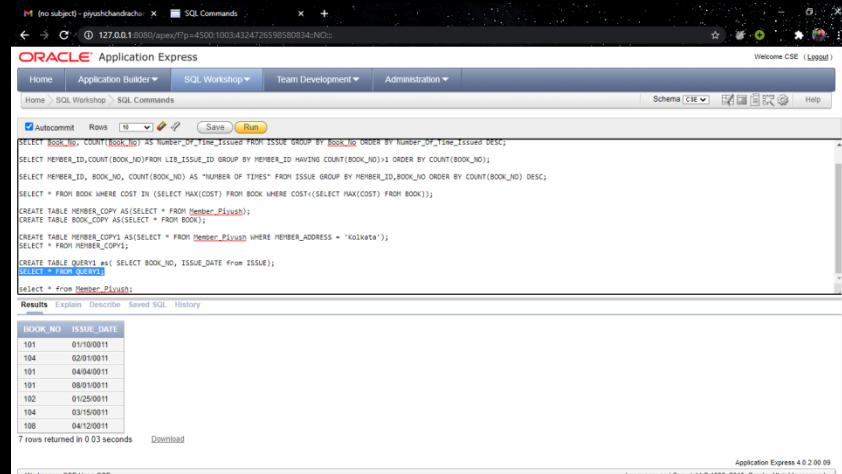
```

The results table shows two rows:

MEMBER_ID	MEMBER_NAME	MEMBER_ADDRESS	ACC_OPEN_DATE	MEMBERSHIP_TYPE	FEES_PAID	MAX_BOOKS_ALLOWED	PENALTY_AMOUNT
6	Suparna Biswas	Kolkata	04/12/2011	Half Yearly	700	1	0
8	Aprita Roy	Kolkata	07/31/2011	Half Yearly	700	1	0

7. Create a table named QUERY1 with two attributes as BOOK_NO, ISSUE_DATE and copy all information about BOOK_NO and ISSUE_DATE from ISSUE table to QUERY1 table.

➤ CREATE TABLE QUERY1 AS(SELECT BOOK_NO, ISSUE_DATE from ISSUE);
 SELECT * FROM QUERY1;



```

SELECT BOOK_NO, COUNT(BOOK_NO) AS Number_of_time_Issued FROM ISSUE GROUP BY BOOK_NO ORDER BY Number_of_time_Issued DESC;
SELECT MEMBER_ID,COUNT(BOOK_NO)FROM L1_ISSUE_ID GROUP BY MEMBER_ID HAVING COUNT(BOOK_NO)>1 ORDER BY COUNT(BOOK_NO);
SELECT MEMBER_ID, BOOK_NO, COUNT(BOOK_NO) AS "NUMBER OF TIMES" FROM ISSUE GROUP BY MEMBER_ID,BOOK_NO ORDER BY COUNT(BOOK_NO) DESC;
SELECT * FROM BOOK WHERE COST < (SELECT MAX(COST) FROM BOOK WHERE COST<(SELECT MAX(COST) FROM BOOK));
CREATE TABLE MEMBER_COPY AS(SELECT * FROM Member_Plivush);
CREATE TABLE BOOK_COPY AS(SELECT * FROM BOOK);
CREATE TABLE MEMBER_COPY1 AS(SELECT * FROM Member_Plivush WHERE MEMBER_ADDRESS = 'Kolkata');
SELECT * FROM MEMBER_COPY1;
CREATE TABLE QUERY1 AS( SELECT BOOK_NO, ISSUE_DATE from ISSUE);
SELECT * FROM QUERY1;
select * from Member_Plivush;
    
```

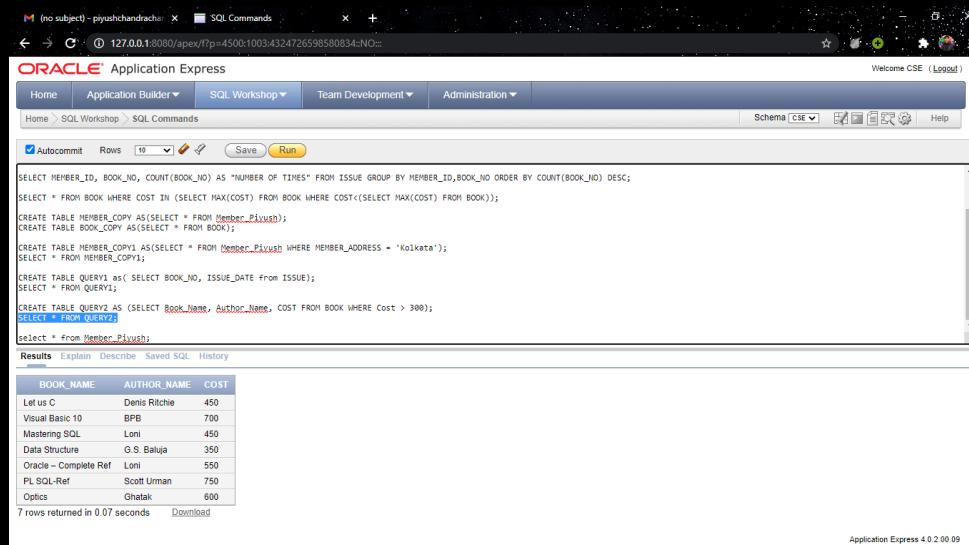
Results

BOOK_NO	ISSUE_DATE
191	01/10/2011
104	02/10/2011
191	04/04/2011
191	08/01/2011
102	01/25/2011
104	03/15/2011
106	04/12/2011

7 rows returned in 0.03 seconds

8. Create and store BOOK_NAME, AUTHOR_NAME and COST to QUERY2 table where cost is greater than 300.

➤ CREATE TABLE QUERY2 AS (SELECT Book_Name, Author_Name, COST FROM BOOK WHERE Cost > 300);
 SELECT * FROM QUERY2;



```

SELECT MEMBER_ID, BOOK_NO, COUNT(BOOK_NO) AS "NUMBER OF TIMES" FROM ISSUE GROUP BY MEMBER_ID,BOOK_NO ORDER BY COUNT(BOOK_NO) DESC;
SELECT * FROM BOOK WHERE COST < (SELECT MAX(COST) FROM BOOK WHERE COST<(SELECT MAX(COST) FROM BOOK));
CREATE TABLE MEMBER_COPY AS(SELECT * FROM Member_Plivush);
CREATE TABLE BOOK_COPY AS(SELECT * FROM BOOK);
CREATE TABLE MEMBER_COPY1 AS(SELECT * FROM Member_Plivush WHERE MEMBER_ADDRESS = 'Kolkata');
SELECT * FROM MEMBER_COPY1;
CREATE TABLE QUERY1 AS( SELECT BOOK_NO, ISSUE_DATE from ISSUE);
SELECT * FROM QUERY1;
CREATE TABLE QUERY2 AS (SELECT Book_Name, Author_Name, COST FROM BOOK WHERE Cost > 300);
SELECT * FROM QUERY2;
select * from Member_Plivush;
    
```

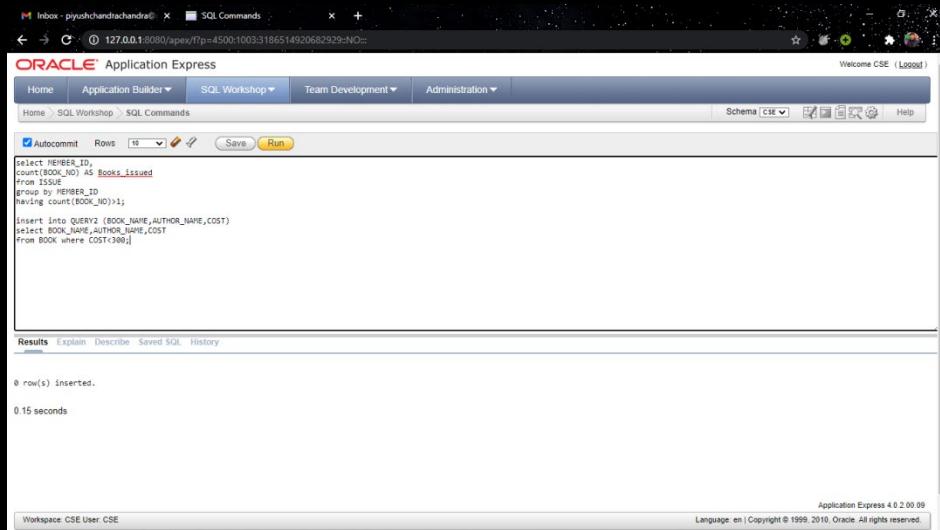
Results

BOOK_NAME	AUTHOR_NAME	COST
Let us C	Denis Ritchie	450
Visual Basic 10	BPB	700
Mastering SQL	Loni	450
Data Structure	G.S. Baluja	350
Oracle - Complete Ref	Loni	550
PLSQL_Ref	Scott Urman	750
Optics	Ghatak	600

7 rows returned in 0.07 seconds

9. Insert BOOK_NAME, AUTHOR_NAME and COST to QUERY2 table where cost is less than 300.

- insert into QUERY2 (BOOK_NAME,AUTHOR_NAME,COST)
select BOOK_NAME,AUTHOR_NAME,COST
from BOOK where COST<300;



```
select MEMBER_ID,  
       count(BOOK_NO) AS Books_Issued  
  from ISSUE  
 group by MEMBER_ID  
 having count(BOOK_NO)>1;  
  
Insert into QUERY2 (BOOK_NAME,AUTHOR_NAME,COST)  
select BOOK_NAME,AUTHOR_NAME,COST  
  from BOOK where COST<300;
```

Results Explain Describe Saved SQL History

0 row(s) inserted.
0.15 seconds

Workspace: CSE User: CSE Application Express 4.0.2.00.09 Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

10. ADD an attribute name AVAILABLE with data type NUMBER (5) in the BOOKS_COPY table and fill up the new attribute with data.

- alter table BOOK_COPY
add available number(5);

```
update BOOK_COPY  
set available = 50  
where BOOK_NO=101;
```

```
update BOOK_COPY  
set available = 100  
where BOOK_NO=102;
```

```
update BOOK_COPY
```

```
set available = 23  
where BOOK_NO=103;
```

```
update BOOK_COPY  
set available = 0  
where BOOK_NO=104;
```

```
update BOOK_COPY  
set available = 88  
where BOOK_NO=105;
```

```
update BOOK_COPY  
set available = 150  
where BOOK_NO=106;
```

```
update BOOK_COPY  
set available = 27  
where BOOK_NO=107;
```

```
update BOOK_COPY  
set available = 90  
where BOOK_NO=108;
```

Inbox - piyushchandrachandra@ 127.0.0.1:8080/apex/r?p=4500:1003:3186514920682929:NO=

Home Application Builder SQL Workshop Team Development Administration

Autocommit Rows 10 Save Run

```

having count(BOOK_NO)>1;
insert into BOOK_COPY(BOOK_NAME,AUTHOR_NAME,COST)
select BOOK_NAME,AUTHOR_NAME,COST
From BOOK where COST<300;
alter table BOOK_COPY
add available number(5);
update BOOK_COPY
set available = 50
where BOOK_NO=101;
update BOOK_COPY
set available = 100
where BOOK_NO=102;
update BOOK_COPY
Results Explain Describe Saved SQL History

```

Table altered.

0.63 seconds

Application Express 4.0.2.0.09

Workspace: CSE User: CSE Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

Inbox - piyushchandrachandra@ 127.0.0.1:8080/apex/r?p=4500:1003:3186514920682929:NO=

Home Application Builder SQL Workshop Team Development Administration

Autocommit Rows 10 Save Run

```

having count(BOOK_NO)>1;
insert into BOOK_COPY(BOOK_NAME,AUTHOR_NAME,COST)
select BOOK_NAME,AUTHOR_NAME,COST
From BOOK where COST<300;
alter table BOOK_COPY
add available number(5);
update BOOK_COPY
set available = 50
where BOOK_NO=101;
update BOOK_COPY
set available = 100
where BOOK_NO=102;
update BOOK_COPY
Results Explain Describe Saved SQL History

```

Table altered.

0.63 seconds

Application Express 4.0.2.0.09

Workspace: CSE User: CSE Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

Inbox - piyushchandrachandra@ 127.0.0.1:8080/apex/r?p=4500:1003:3186514920682929:NO=

Home Application Builder SQL Workshop Team Development Administration

Autocommit Rows 10 Save Run

```

update BOOK_COPY
set available = 88
where BOOK_NO=105;
update BOOK_COPY
set available = 150
where BOOK_NO=106;
update BOOK_COPY
set available = 27
where BOOK_NO=107;
update BOOK_COPY
set available = 90
where BOOK_NO=108;
SELECT * FROM BOOK_COPY;

```

Results Explain Describe Saved SQL History

BOOK_NO	BOOK_NAME	AUTHOR_NAME	COST	CATEGORY	AVAILABLE
101	Let us C	Denis Ritchie	450	Others	50
103	Visual Basic 10	BPB	700	Others	23
104	Mastering SQL	Loni	450	Database	0
106	UNIX	Sumitava Das	300	System	150
108	Data Structure	G.S. Baluja	350	Others	90
102	Oracle – Complete Ref	Loni	550	Database	100
105	PL SQL Ref	Scott Urman	750	Database	88
107	Optics	Ghatak	600	Science	27

8 rows returned in 0.00 seconds Download

Application Express 4.0.2.0.09

Workspace: CSE User: CSE Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

10. Change the Data type of CATEGORY attribute to VARCHAR2(15) in BOOKS_COPY table.

➤ alter table BOOK_COPY
modify(CATEGORY VARCHAR2(15));
desc BOOK_COPY;

The screenshot shows the Oracle Application Express SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL code:

```
update BOOK_COPY
set available = 150
where BOOK_NO=108;
update BOOK_COPY
set available = 27
where BOOK_NO=107;
update BOOK_COPY
set available = 98
where BOOK_NO=108;
SELECT * FROM BOOK_COPY;
alter table BOOK_COPY
modify( CATEGORY VARCHAR2(15));
desc BOOK_COPY;
```

Below the code, the Results tab is open, showing the structure of the BOOK_COPY table:

Object Type	Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
Table	BOOK_COPY	BOOK_NO	NUMBER	-	6	0	-	✓	-	-
		BOOK_NAME	VARCHAR2	30	-	-	-	-	-	-
		AUTHOR_NAME	VARCHAR2	30	-	-	-	✓	-	-
		COST	NUMBER	-	7	2	-	✓	-	-
		CATEGORY	VARCHAR2	15	-	-	-	✓	-	-
		AVAILABLE	NUMBER	-	5	0	-	✓	-	-

The results pane shows the output of the SELECT * query, which is identical to the table structure shown above. The status bar at the bottom indicates "Application Express 4.0.2.00.09" and "Language: en | Copyright © 1999, 2010, Oracle. All rights reserved."

11. Create a SAVE POINT after that do some delete or update operation and roll back the previous scenario.

➤ CREATE TABLE STUDENT(
ROLL NUMBER(10),
NAME VARCHAR(30)
);
INSERT INTO STUDENT VALUES(1,'PIYUSH');
SAVEPOINT A;
INSERT INTO STUDENT VALUES(2,'SARIT');
SAVEPOINT B;
INSERT INTO STUDENT VALUES(3,'TANMOY');
SAVEPOINT C;
INSERT INTO STUDENT VALUES(4,'SMARTO');
SAVEPOINT D;
SELECT * FROM STUDENT

```
ROLLBACK TO C;  
SELECT * FROM STUDENT;  
ROLLBACK TO B;  
SELECT * FROM STUDENT;
```

```
SQL> INSERT INTO BOOKS VALUES(108,'ABC Chemistry','Dr SP Jauhar',600,'Science');  
1 row created.  
SQL> COMMIT;  
Commit complete.  
SQL> UPDATE BOOKS SET AUTHOR_NAME ='Dr Theodore Roosevelt' WHERE BOOK_NO ='108';  
1 row updated.  
SQL> SAVEPOINT A;  
Savepoint created.  
SQL> INSERT INTO BOOKS VALUES(109,'Database Systems','Peter Rob',150,'Database');  
1 row created.  
SQL> SAVEPOINT B;  
Savepoint created.  
SQL> INSERT INTO BOOKS VALUES(110,'ETHICAL HACKING','Peter Kim',500,'Technology');  
1 row created.  
SQL> SAVEPOINT C;  
Savepoint created.  
SQL> SELECT * FROM BOOKS;  
BOOK_NO BOOK_NAME AUTHOR_NAME  
-----  
COST CATEGORY  
-----  
108 ABC Chemistry Dr Theodore Roosevelt  
600 Science  
  
109 Database Systems Peter Rob  
150 Database  
  
110 ETHICAL HACKING Peter Kim  
500 Technology  
SQL> ROLLBACK TO B;  
Rollback complete.  
SQL> SELECT * FROM BOOKS;  
BOOK_NO BOOK_NAME AUTHOR_NAME  
-----  
COST CATEGORY  
-----  
108 ABC Chemistry Dr Theodore Roosevelt  
600 Science  
  
109 Database Systems Peter Rob  
150 Database
```

12.Create a New USER, Log in to new USER; provide permission of selection on BOOK table to the new USER. Then provide DBA privilege to the new USER.

➤ CREATE USER U1(ANY USER NAME) IDENTIFIED BY " ORACLE FIRST LOGIN PASSWORD";
>USER CREATED
GRANT CREATE SESSION TO U1;
>GRANT SUCCEEDED
GRANT DBA TO U1;
>GRANT SUCCEEDED
SELECT USERNAME FROM DBA_USERS;
U1> WILL BE HIGHLITED

```
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\USER>sqlplus/ as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on Wed Dec 2 20:53:16 2020

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - Production

SQL> CREATE USER U1 IDENTIFIED BY "system";

User created.

SQL> grant create session to U1;

Grant succeeded.

SQL> GRANT DBA TO U1;

Grant succeeded.
```

```
SQL> SELECT USERNAME FROM DBA_USERS;

USERNAME
-----
SYSTEM
SYS
U1
USER1
ANONYMOUS
CSE
APEX_PUBLIC_USER
APEX_040000
FLOWS_FILES
XS$NULL
OUTLN

,USERNAME
-----
XDB
CTXSYS
MDSYS
HR
```

```
GRANT SELECT ON BOOKS TO U1;
```

Results Explain Describe Saved S

Statement processed.

0.00 seconds

ASSIGNMENT 6

18/11/2020 – 24/11/2020

1. (A)

1. Create the following tables with appropriate constraints using SQL command.

Example:

A) Table Name: Worker

COLUMN NAME	DATA TYPE	DESCRIPTION
Worker_Id	Number(5)	Unique worker ID
First_Name	Varchar2(30)	Name of the worker
Last_Name	Varchar2(30)	Last Name of the worker
Salary	Number(7)	Salary paid to the worker.
Joining_Date	Date	Date of joining of workers
Department	Varchar2(6)	Type of the Department such as 'HR', 'Admin', 'Account'.

CONSTRAINT:

- a. Worker_Id – Primary Key, Auto Increment.
- b. First_Name – NOT NULL
- c. Department - 'HR', 'Admin', 'Account'.

➤ create table WORKER

(

Worker_Id number(5) PRIMARY KEY,

First_Name varchar2(30) NOT NULL,

Last_Name varchar2(30),

Salary number(7),

Joining_Date date,

Department varchar2(6)

);

ALTER TABLE WORKER MODIFY Department VARCHAR2(7);

The screenshot shows the Oracle Application Express interface. In the top navigation bar, there are tabs for Home, Application Builder, SQL Workshop (selected), Team Development, and Administration. Below the navigation bar, the path is Home > SQL Workshop > SQL Commands. The main area contains the following SQL code:

```

INSERT INTO WORKER VALUES(001, 'Monika', 'Arona', 100000, '02-20-14', 'HR');
INSERT INTO WORKER VALUES(002, 'Nihariks', 'Verma', 80000, '06-11-14', 'Admin');
INSERT INTO WORKER VALUES(003, 'Vishal', 'Singhal', 300000, '02-20-14', 'HR');
INSERT INTO WORKER VALUES(004, 'Amitabh', 'Singh', 500000, '02-20-14', 'Admin');
INSERT INTO WORKER VALUES(005, 'Vivek', 'Bhati', 500000, '06-11-14', 'Admin');
INSERT INTO WORKER VALUES(006, 'Vipul', 'Diwan', 200000, '06-11-14', 'Account');
INSERT INTO WORKER VALUES(007, 'Satish', 'Kumar', 75000, '01-20-14', 'Account');
INSERT INTO WORKER VALUES(008, 'Geetika', 'Chauhan', 90000, '04-11-14', 'Admin');

DESC WORKER;
SELECT * FROM WORKER;

ALTER TABLE WORKER MODIFY Department VARCHAR2( 7 );

```

Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. A table titled "Object Type TABLE Object WORKER" is displayed, showing the structure of the WORKER table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WORKER	WORKER_ID	NUMBER	-	5	0	1	-	-	-
	FIRST_NAME	VARCHAR2	30	-	-	-	-	-	-
	LAST_NAME	VARCHAR2	30	-	-	-	✓	-	-
	SALARY	NUMBER	-	7	0	-	✓	-	-
	JOINING_DATE	DATE	7	-	-	-	✓	-	-
	DEPARTMENT	VARCHAR2	7	-	-	-	✓	-	-

1 - 6

(B)

B) Table Name: Bonus

COLUMN NAME	DATA TYPE	DESCRIPTION
Worker_Ref_Id	Number(5)	Worker identification number
Bonus_Date	Date	Bonus Date of the Worker
Bonus_Amount	Number(7)	Bonus Amount of the Worker

CONSTRAINT:

- a. Worker_Ref_Id – Foreign Key.
- b. Bonus_Amount – Not Null

➤ CREATE TABLE BONUS (

```

WORKER_REF_ID number(5),
BONUS_AMOUNT number(5) NOT NULL,
BONUS_DATE date
);

```

```

SQL Commands
ORACLE Application Express
Home Application Builder SQL Workshop Team Development Administration
Welcome CSE (L2092d)
Home SQL Workshop SQL Commands
AutoCommit Rows 10 Save Run
INSERT INTO WORKER VALUES(001, 'Vishal', 'Dewan', 200000, '06-11-14', 'Account');
INSERT INTO WORKER VALUES(002, 'Sekhsih', 'Kumar', 75000, '01-08-14', 'Account');
INSERT INTO WORKER VALUES(003, 'Gestika', 'Chauhan', 90000, '04-11-14', 'Admin');
DESC WORKER;
SELECT * FROM WORKER;
DESC BONUS;
ALTER TABLE WORKER MODIFY Department VARCHAR2( 7 );

INSERT INTO BONUS VALUES(001, 5000, '02-20-16');
INSERT INTO BONUS VALUES(002, 4000, '02-20-16');
INSERT INTO BONUS VALUES(003, 4000, '02-20-16');
INSERT INTO BONUS VALUES(001, 4500, '02-20-16');
INSERT INTO BONUS VALUES(002, 3500, '06-11-16');
SELECT * FROM BONUS;
DESC BONUS;

Results Explain Describe Saved SQL History
Object Type: TABLE Object: BONUS
Table Column Data Type Length Precision Scale Primary Key Nullable Default Comment
BONUS WORKER_REF_ID NUMBER - 5 0 - ✓ - -
BONUS_AMOUNT NUMBER - 5 0 - - - -
BONUS_DATE DATE 7 - - - ✓ - -
1 - 3

```

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

(C)

C) Table Name : Title

COLUMN NAME	DATA TYPE	DESCRIPTION
Worker_Ref_Id	Number(5)	Worker identification number
Worker_Title	Varchar2(30)	Worker Title used for post.
Affected_From	Date	Date of joining of workers

CONSTRAINT:

- a. Worker_Ref_Id -Foreign Key.
- b. Worker_Title – Not Null.
- c. Worker_Title – ‘Manager’/‘Lead’/‘Asst. Manager’/‘Executive’.

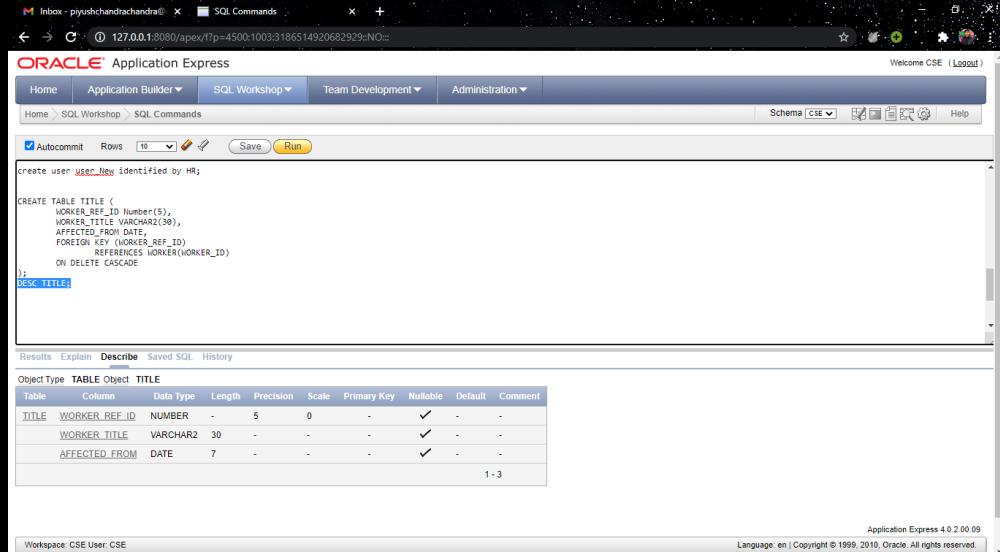
➤ CREATE TABLE TITLE (

```

WORKER_REF_ID Number(5),
WORKER_TITLE VARCHAR2(30),
AFFECTED_FROM DATE,
FOREIGN KEY (WORKER_REF_ID)

```

REFERENCES WORKER(WORKER_ID)
 ON DELETE CASCADE
);



The screenshot shows the Oracle Application Express SQL Workshop interface. The code entered is:

```

create user user_New identified by HR;

CREATE TABLE TITLE (
  WORKER_REF_ID Number(5),
  WORKER_TITLE VARCHAR2(30),
  AFFECTED_FROM DATE,
  FOREIGN KEY (WORKER_REF_ID)
    REFERENCES WORKER(WORKER_ID)
  ON DELETE CASCADE
);
DESC TITLE;
  
```

The results pane shows the table structure:

Object Type	Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TABLE	TITLE	WORKER_REF_ID	NUMBER	-	5	0	-	✓	-	-
		WORKER_TITLE	VARCHAR2	30	-	-	-	✓	-	-
		AFFECTED_FROM	DATE	7	-	-	-	✓	-	-

1 - 3

2. (A)

A) Table Name: Worker

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	2014-02-20	HR
2	Niharika	Verma	80000	2014-06-11	Admin
3	Vishal	Singhal	300000	2014-02-20	HR
4	Amitabh	Singh	500000	2014-02-20	Admin
5	Vivek	Bhati	500000	2014-06-11	Admin
6	Vipul	Diwan	200000	2014-06-11	Account
7	Satish	Kumar	75000	2014-01-20	Account
8	Geetika	Chauhan	90000	2014-04-11	Admin

- INSERT INTO WORKER VALUES(001, 'Monika', 'Arora', 100000, '02-20-14', 'HR');
- INSERT INTO WORKER VALUES(002, 'Niharika', 'Verma', 80000, '06-11-14', 'Admin');
- INSERT INTO WORKER VALUES(003, 'Vishal', 'Singhal', 300000, '02-20-14', 'HR');
- INSERT INTO WORKER VALUES(004, 'Amitabh', 'Singh', 500000, '02-20-14', 'Admin');
- INSERT INTO WORKER VALUES(005, 'Vivek', 'Bhati', 500000, '06-11-14', 'Admin');
- INSERT INTO WORKER VALUES(006, 'Vipul', 'Diwan', 200000, '06-11-14', 'Account');

```

INSERT INTO WORKER VALUES(007, 'Satish', 'Kumar', 75000, '01-20-14', 'Account' );

INSERT INTO WORKER VALUES(008, 'Geetika', 'Chauhan', 90000, '04-11-14', 'Admin' );

DESC WORKER;

SELECT * FROM WORKER;

```

The screenshot shows the Oracle Application Express interface with the SQL Workshop tab selected. The SQL editor contains the following code:

```

INSERT INTO WORKER VALUES(001, 'Monika', 'Arora', 100000, '02-20-14', 'HR');
INSERT INTO WORKER VALUES(002, 'Niharika', 'Verma', 80000, '06-11-14', 'Admin');
INSERT INTO WORKER VALUES(003, 'Vishal', 'Singhal', 300000, '02-20-14', 'HR');
INSERT INTO WORKER VALUES(004, 'Amitabh', 'Singh', 500000, '02-20-14', 'Admin');
INSERT INTO WORKER VALUES(005, 'Vivek', 'Bhati', 500000, '06-11-14', 'Admin');
INSERT INTO WORKER VALUES(006, 'Vipul', 'Divan', 200000, '06-11-14', 'Account');
INSERT INTO WORKER VALUES(007, 'Satish', 'Kumar', 75000, '01-20-14', 'Account');
INSERT INTO WORKER VALUES(008, 'Geetika', 'Chauhan', 90000, '04-11-14', 'Admin');

DESC WORKER;

SELECT * FROM WORKER;

ALTER TABLE WORKER MODIFY Department VARCHAR2( 7 );

```

The results section displays the data from the WORKER table:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	02/20/0014	HR
2	Niharika	Verma	80000	06/11/0014	Admin
3	Vishal	Singhal	300000	02/20/0014	HR
5	Vivek	Bhati	500000	06/11/0014	Admin
8	Geetika	Chauhan	90000	04/11/0014	Admin
4	Amitabh	Singh	500000	02/20/0014	Admin
6	Vipul	Divan	200000	06/11/0014	Account
7	Satish	Kumar	75000	01/20/0014	Account

8 rows returned in 0.01 seconds [Download](#)

(B)

B) Table Name: Bonus

WORKER_REF_ID	BONUS_DATE	BONUS_AMOUNT
1	2016-02-20	5000
2	2016-06-11	3000
3	2016-02-20	4000
1	2016-02-20	4500
2	2016-06-11	3500

- INSERT INTO BONUS VALUES(001, 5000, '02-20-16');
 - INSERT INTO BONUS VALUES(002, 3000, '06-11-16');
 - INSERT INTO BONUS VALUES(003, 4000, '02-20-16');
 - INSERT INTO BONUS VALUES(001, 4500, '02-20-16');
 - INSERT INTO BONUS VALUES(002, 3500, '06-11-16');
- SELECT * FROM BONUS;

The screenshot shows the Oracle Application Express interface. In the top navigation bar, 'SQL Commands' is selected. Below it, the 'SQL Workshop' tab is active. The main area contains several SQL statements:

```

INSERT INTO WORKER VALUES(006, 'Vipul', 'Dikken', 200000, '05-11-14', 'Account');
INSERT INTO WORKER VALUES(007, 'Satish', 'Kumar', 75000, '01-20-14', 'Accounts');
INSERT INTO WORKER VALUES(008, 'Geetika', 'Chauhan', 90000, '04-11-14', 'Admin');
SELECT * FROM WORKER;
ALTER TABLE WORKER MODIFY Department VARCHAR2( 7 );

INSERT INTO BONUS VALUES(001, 5000, '02-20-16' );
INSERT INTO BONUS VALUES(002, 3000, '06-11-16' );
INSERT INTO BONUS VALUES(003, 4000, '02-20-16' );
INSERT INTO BONUS VALUES(004, 4500, '02-20-16' );
INSERT INTO BONUS VALUES(005, 3500, '06-11-16' );
SELECT * FROM BONUS;

```

Below the code, the 'Results' tab is selected, showing the output of the SELECT statement:

WORKER_REF_ID	BONUS_AMOUNT	BONUS_DATE
1	5000	02/20/2016
2	3000	06/11/2016
3	4000	02/20/2016
1	4500	02/20/2016
2	3500	06/11/2016

At the bottom right, it says '5 rows returned in 0.04 seconds'.

(B)

C) Table Name: Title

WORKER_REF_ID	WORKER_TITLE	AFFECTED_FROM
1	Manager	2016-02-20
2	Executive	2016-06-11
8	Executive	2016-06-11
5	Manager	2016-06-11
4	Asst. Manager	2016-06-11
7	Executive	2016-06-11
6	Lead	2016-06-11
3	Lead	2016-06-11

- INSERT INTO TITLE VALUES(1, 'Manager', '02-20-16');
- INSERT INTO TITLE VALUES(2, 'Executive', '06-11-16');
- INSERT INTO TITLE VALUES(8, 'Executive', '06-11-16');
- INSERT INTO TITLE VALUES(5, 'Manager', '06-11-16');
- INSERT INTO TITLE VALUES(4, 'Asst. Manager', '06-11-16');
- INSERT INTO TITLE VALUES(7, 'Executive', '06-11-16');

```

INSERT INTO TITLE VALUES(6, 'Lead', '06-11-16');

INSERT INTO TITLE VALUES(3, 'Lead', '06-11-16');

SELECT * FROM TITLE;

```

The screenshot shows the Oracle Application Express interface with the SQL Workshop tab selected. The code area contains the following SQL:

```

;/
DESC TITLE;

INSERT INTO TITLE VALUES(1, 'Manager', '06-11-16');
INSERT INTO TITLE VALUES(2, 'Executive', '06-11-16');
INSERT INTO TITLE VALUES(3, 'Lead', '06-11-16');
INSERT INTO TITLE VALUES(4, 'Asst. Manager', '06-11-16');
INSERT INTO TITLE VALUES(5, 'Executive', '06-11-16');
INSERT INTO TITLE VALUES(6, 'Lead', '06-11-16');
SELECT * FROM TITLE;

```

The results section displays the following table:

WORKER_REF_ID	WORKER_TITLE	AFFECTED_FROM
1	Manager	02/00/016
2	Executive	06/11/0016
3	Executive	06/11/0016
5	Manager	06/11/0016
4	Asst. Manager	06/11/0016
7	Executive	06/11/0016
6	Lead	06/11/0016
3	Lead	06/11/0016

8 rows returned in 0.18 seconds Download

1. Write An SQL Query To Fetch “FIRST_NAME” From Worker Table Using The Alias Name WORKER_NAME.
- Select FIRST_NAME AS WORKER_NAME from WORKER;

The screenshot shows the Oracle Application Express interface with the SQL Workshop tab selected. The code area contains the following SQL:

```

;/
DESC TITLE;

INSERT INTO TITLE VALUES(1, 'Executive', '06-11-16');
INSERT INTO TITLE VALUES(6, 'Lead', '06-11-16');
INSERT INTO TITLE VALUES(3, 'Lead', '06-11-16');
SELECT * FROM TITLE;

Select FIRST_NAME AS WORKER_NAME from WORKER;
Select upper(FIRST_NAME) from Worker;
Select distinct DEPARTMENT from Worker;
Select substring(FIRST_NAME,1,3) from Worker;
Select INSTR(FIRST_NAME, 'BINARY') from Worker where FIRST_NAME = 'Amitabh';
Select RTRIM(FIRST_NAME) from Worker;

```

The results section displays the following table:

WORKER_NAME
Monika
Niharika
Vishal
Vivek
Geetika
Amitabh
Vipul
Satish

8 rows returned in 0.07 seconds Download

2. Write An SQL Query To Fetch “FIRST_NAME” From Worker Table In Upper Case.

- Select upper(FIRST_NAME) from WORKER;

The screenshot shows the Oracle Application Express interface with the SQL Commands page open. The query entered is:

```
INSERT INTO TITLE VALUES(7, 'Executive', '06-11-16');
INSERT INTO TITLE VALUES(6, 'Lead', '06-11-16');
INSERT INTO TITLE VALUES(5, 'Lead', '06-11-16');
SELECT * FROM TITLE;

Select FIRST_NAME AS WORKER_NAME from WORKER;
Select upper(FIRST_NAME) from WORKER;
Select distinct DEPARTMENT from WORKER;
Select substr(FIRST_NAME,1,3) from WORKER;
Select INSTR(FIRST_NAME, BINARY'a') from WORKER where FIRST_NAME = 'Amitabh';
Select RTRIM(FIRST_NAME) from WORKER;
```

The results section shows the output of the `upper(FIRST_NAME)` query:

UPPER(FIRST_NAME)
MONIKA
NIHARIKA
VISHAL
VIVEK
GEETIKA
AMITABH
VIPUL
SATISH

8 rows returned in 0.00 seconds

3. Write An SQL Query To Fetch Unique Values Of DEPARTMENT From Worker Table.

- Select distinct DEPARTMENT from WORKER;

The screenshot shows the Oracle Application Express interface with the SQL Workshop page open. The query entered is identical to the one in the previous screenshot:

```
INSERT INTO TITLE VALUES(7, 'Executive', '06-11-16');
INSERT INTO TITLE VALUES(6, 'Lead', '06-11-16');
INSERT INTO TITLE VALUES(5, 'Lead', '06-11-16');
SELECT * FROM TITLE;

Select FIRST_NAME AS WORKER_NAME from WORKER;
Select upper(FIRST_NAME) from WORKER;
Select distinct DEPARTMENT from WORKER;
Select substr(FIRST_NAME,1,3) from WORKER;
Select INSTR(FIRST_NAME, BINARY'a') from WORKER where FIRST_NAME = 'Amitabh';
Select RTRIM(FIRST_NAME) from WORKER;
```

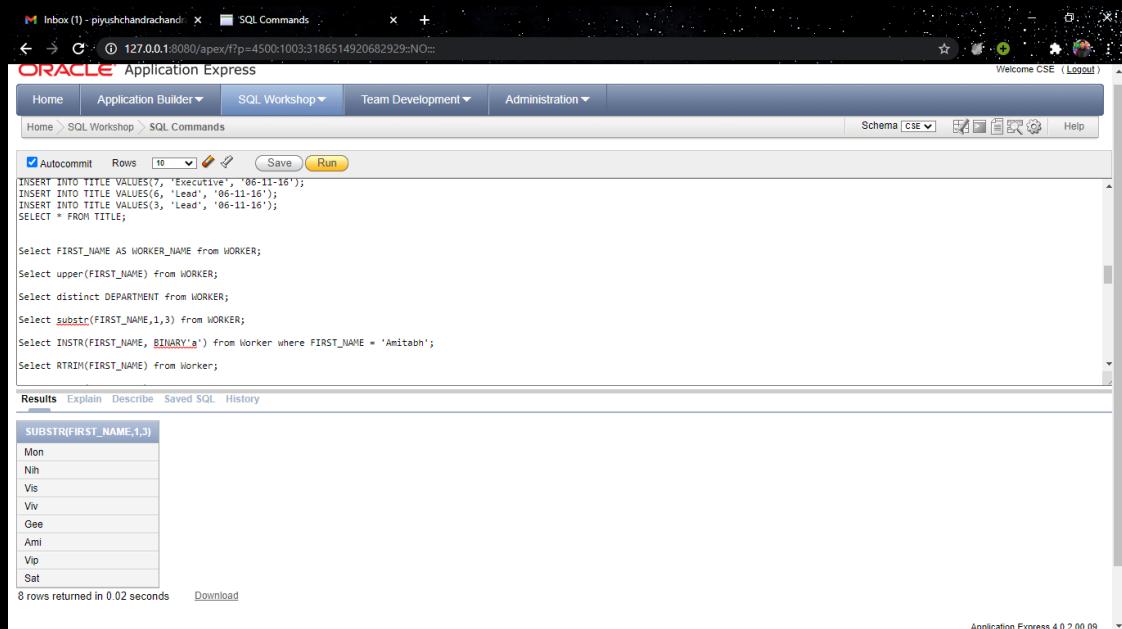
The results section shows the output of the `distinct DEPARTMENT` query:

DEPARTMENT
HR
Account
Admin

3 rows returned in 0.06 seconds

4. Write an SQL Query to Print the First Three Characters Of FIRST_NAME from Worker Table.

- Select substr(FIRST_NAME,1,3) from WORKER;



The screenshot shows the Oracle Application Express SQL Workshop interface. The SQL editor contains the following code:

```
INSERT INTO TITLE VALUES(7, 'Executive', '06-11-16');
INSERT INTO TITLE VALUES(6, 'Lead', '06-11-16');
INSERT INTO TITLE VALUES(3, 'Lead', '06-11-16');
SELECT * FROM TITLE;

Select FIRST_NAME AS WORKER_NAME from WORKER;
Select upper(FIRST_NAME) from WORKER;
Select distinct DEPARTMENT from WORKER;
Select substr(FIRST_NAME,1,3) from WORKER;
Select INSTR(FIRST_NAME, BINARY'a') from WORKER where FIRST_NAME = 'Amitabh';
Select RTRIM(FIRST_NAME) from WORKER;
```

The results section displays the output of the last query, which is:

SUBSTR(FIRST_NAME,1,3)
Mon
Nih
Vis
Viv
Gee
Ami
Vip
Sat

8 rows returned in 0.02 seconds

5. Write an SQL Query To Find The Position Of The Alphabet („A“) In The First Name Column „Amitabh“ From Worker Table.

- Select INSTR(FIRST_NAME,'A') from WORKER where FIRST_NAME = 'Amitabh';

The screenshot shows the Oracle Application Express interface. In the SQL Commands workspace, the following SQL code is run:

```
Select INSTR(FIRST_NAME,'A') from WORKER where FIRST_NAME = 'Amitabh';
```

The results pane displays the output:

INSTR(FIRST_NAME,'A')
1

1 rows returned in 0.19 seconds

6. Write An SQL Query To Print The FIRST_NAME From Worker Table After Removing White Spaces From The Right Side.

➤ Select RTRIM(FIRST_NAME) from WORKER;

The screenshot shows the Oracle Application Express interface. In the SQL Workshop workspace, the following SQL code is run:

```
Select upper(FIRST_NAME) from WORKER;
Select distinct DEPARTMENT from WORKER;
Select substr(FIRST_NAME,1,3) from WORKER;
Select INSTR(FIRST_NAME, BINARY 'a') from WORKER where FIRST_NAME = 'Amitabh';
Select RTRIM(FIRST_NAME) from WORKER;
Select LTRIM(DEPARTMENT) from WORKER;
Select distinct length(DEPARTMENT) from WORKER;
Select REPLACE(FIRST_NAME,' ','') from WORKER;
Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' from WORKER;
```

The results pane displays the output:

RTRIM(FIRST_NAME)
Monika
Niharika
Vishal
Vivek
Geetika
Amitabh
Vipul
Satish

8 rows returned in 0.03 seconds

7. Write An SQL Query To Print The DEPARTMENT From Worker Table After Removing White Spaces From The Left Side.

➤ Select LTRIM(DEPARTMENT) from WORKER;

The screenshot shows the Oracle Application Express interface with the following details:

- URL:** 127.0.0.1:8080/apex/r?p=4500:1003:3186514920682929:NO::
- Tab:** SQL Commands
- SQL Query:**

```
SELECT upper(FIRST_NAME) FROM WORKER;
SELECT distinct DEPARTMENT FROM WORKER;
SELECT substr(FIRST_NAME,1,3) FROM WORKER;
SELECT INSTR(FIRST_NAME, BINARY 'a') FROM WORKER WHERE FIRST_NAME = 'Amitabh';
SELECT RTRIM(FIRST_NAME) FROM WORKER;
SELECT LTRIM(DEPARTMENT) FROM WORKER;
SELECT distinct length(DEPARTMENT) FROM WORKER;
SELECT REPLACE(FIRST_NAME,'a','A') FROM WORKER;
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' FROM WORKERS;
```

- Results:** A table titled "LTRIM(DEPARTMENT)" showing the following data:

LTRIM(DEPARTMENT)
HR
Admin
HR
Admin
Admin
Admin
Account
Account

8 rows returned in 0.01 seconds

8. Write An SQL Query That Fetches The Unique Values Of DEPARTMENT From Worker Table And Prints Its Length.

➤ Select distinct length(DEPARTMENT) from Worker;

The screenshot shows the Oracle Application Express interface with the following details:

- URL:** 127.0.0.1:8080/apex/r?p=4500:1003:3186514920682929:NO::
- Tab:** SQL Commands
- SQL Query:**

```
SELECT upper(FIRST_NAME) FROM WORKER;
SELECT distinct DEPARTMENT FROM WORKER;
SELECT substr(FIRST_NAME,1,3) FROM WORKER;
SELECT INSTR(FIRST_NAME, BINARY 'a') FROM WORKER WHERE FIRST_NAME = 'Amitabh';
SELECT RTRIM(FIRST_NAME) FROM WORKER;
SELECT LTRIM(DEPARTMENT) FROM WORKER;
SELECT distinct length(DEPARTMENT) FROM WORKER;
SELECT REPLACE(FIRST_NAME,'a','A') FROM WORKER;
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' FROM WORKERS;
```

- Results:** A table titled "LENGTH(DEPARTMENT)" showing the following data:

LENGTH(DEPARTMENT)
2
5
7

3 rows returned in 0.00 seconds

9. Write An SQL Query To Print The FIRST_NAME From Worker Table After Replacing „A“ With „A“.

➤ Select REPLACE(FIRST_NAME,'a','A') from Worker;

The screenshot shows the Oracle Application Express interface. In the top navigation bar, the URL is 127.0.0.1:8080/apex/f?p=4500:1003:3186514920692929::NO:::. The main content area displays several SQL queries:

```

Select upper(FIRST_NAME) from WORKER;
Select distinct DEPARTMENT from WORKER;
Select substr(FIRST_NAME,1,3) from WORKER;
Select INSTR(FIRST_NAME, BINARY 'a') from WORKER where FIRST_NAME = 'Amitabh';
Select RTRIM(FIRST_NAME) from WORKER;
Select LTRIM(DEPARTMENT) from WORKER;
Select distinct length(DEPARTMENT) from WORKER;
Select REPLACE(FIRST_NAME,'a','A') from WORKER;
Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' from WORKER;

```

The results section shows the output of the REPLACE query:

REPLACE(FIRST_NAME,'a','A')
Monika
Niharika
Vishal
Vivek
Geetika
Amitabh
Vipul
Satish

8 rows returned in 0.01 seconds

10. Write An SQL Query To Print The FIRST_NAME And LAST_NAME From Worker Table Into A Single Column COMPLETE_NAME. A Space Char Should Separate Them.

- Select CONCAT(CONCAT(FIRST_NAME, ' '),LAST_NAME)AS "COMPLETE_NAME" FROM WORKER;

The screenshot shows the Oracle Application Express interface. In the top navigation bar, the URL is 127.0.0.1:8080/apex/f?p=4500:1003:2571053800063041::NO:::. The main content area displays the following SQL query:

```

Select INSTR(FIRST_NAME,'A') from WORKER where FIRST_NAME = 'Amitabh';

Select CONCAT(CONCAT(FIRST_NAME, ' '),LAST_NAME)AS "COMPLETE_NAME" FROM WORKER;

```

The results section shows the output of the final query:

COMPLETE_NAME
Monika Arora
Niharika Verma
Vishal Singh
Vivek Bhati
Geetika Chauhan
Amitabh Singh
Vipul Diwan
Satish Kumar

8 rows returned in 0.02 seconds

11. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST_NAME Ascending.

- Select * from Worker order by FIRST_NAME asc;

The screenshot shows the Oracle Application Express interface. The SQL Workshop tab is selected. The code entered is:

```

Select LTRIM(FIRST_NAME, ' ') FROM WORKER WHERE FIRST_NAME = 'Amitabh';
Select RTRIM(FIRST_NAME) from WORKER;
Select LTRIM(DEPARTMENT) from WORKER;
Select distinct length(DEPARTMENT) from WORKER;
Select REPLACE(FIRST_NAME,'a','A') from WORKER;
Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' from WORKER;
Select * from Worker order by FIRST_NAME asc;
Select * from Worker order by FIRST_NAME desc;
Select * from Worker where FIRST_NAME in ('Vipul','Satish');

```

The results section displays a table with the following data:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
4	Amitabh	Singh	500000	02/20/0014	Admin
8	Geetika	Chauhan	90000	04/11/0014	Admin
1	Monika	Arora	100000	02/20/0014	HR
2	Niharika	Verma	80000	06/11/0014	Admin
7	Satish	Kumar	75000	01/20/0014	Account
6	Vipul	Divan	200000	06/11/0014	Account
3	Vishal	Singhal	300000	02/20/0014	HR
5	Vivek	Bhati	500000	06/11/0014	Admin

8 rows returned in 0.04 seconds [Download](#)

Application Express 4.0.2.00.09

12. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST_NAME Ascending And DEPARTMENT Descending.

➤ Select * from Worker order by FIRST_NAME asc,DEPARTMENT desc;

The screenshot shows the Oracle Application Express interface. The SQL Workshop tab is selected. The code entered is identical to the one in the previous screenshot:

```

Select LTRIM(FIRST_NAME, ' ') FROM WORKER WHERE FIRST_NAME = 'Amitabh';
Select RTRIM(FIRST_NAME) from WORKER;
Select LTRIM(DEPARTMENT) from WORKER;
Select distinct length(DEPARTMENT) from WORKER;
Select REPLACE(FIRST_NAME,'a','A') from WORKER;
Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' from WORKER;
Select * from Worker order by FIRST_NAME asc;
Select * from Worker order by FIRST_NAME desc;
Select * from Worker where FIRST_NAME in ('Vipul','Satish');

```

The results section displays a table with the following data:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
4	Amitabh	Singh	500000	02/20/0014	Admin
8	Geetika	Chauhan	90000	04/11/0014	Admin
1	Monika	Arora	100000	02/20/0014	HR
2	Niharika	Verma	80000	06/11/0014	Admin
7	Satish	Kumar	75000	01/20/0014	Account
6	Vipul	Divan	200000	06/11/0014	Account
3	Vishal	Singhal	300000	02/20/0014	HR
5	Vivek	Bhati	500000	06/11/0014	Admin

8 rows returned in 0.00 seconds [Download](#)

Application Express 4.0.2.00.09

13. Write An SQL Query To Print Details For Workers With The First Name As “Vipul” And “Satish” From Worker Table.

➤ Select * from Worker where FIRST_NAME in ('Vipul','Satish');

The screenshot shows the Oracle Application Express interface. In the top navigation bar, there are tabs for Home, Application Builder, SQL Workshop (which is selected), Team Development, and Administration. Below the navigation bar, the URL is 127.0.0.1:8080/apex/f?p=4500:1003:3186514920682929::NO:: and the page title is ORACLE Application Express.

In the main content area, there is a SQL editor window with the following SQL code:

```

SELECT FIRST_NAME, LAST_NAME FROM WORKER WHERE FIRST_NAME = 'Admin';
Select RTRIM(FIRST_NAME) FROM WORKER;
Select LTRIM(DEPARTMENT) FROM WORKER;
Select distinct length(DEPARTMENT) FROM WORKER;
Select REPLACE(FIRST_NAME,'a','A') FROM WORKER;
Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' FROM WORKER;
Select * FROM WORKER ORDER BY FIRST_NAME ASC;
Select * FROM WORKER ORDER BY FIRST_NAME ASC, DEPARTMENT DESC;
Select * FROM WORKER WHERE FIRST_NAME IN ('Vipul','Satish');

```

Below the SQL editor, there is a results grid showing two rows of data:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
6	Vipul	Diwan	200000	06/11/0014	Account
7	Satish	Kumar	75000	01/20/0014	Account

Text at the bottom of the results grid: 2 rows returned in 0.05 seconds [Download](#)

Bottom right corner: Application Express 4.0.2.00.09 Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

14. Write An SQL Query To Print Details Of Workers Excluding First Names, “Vipul” And “Satish” From Worker Table.

- Select * from Worker where FIRST_NAME not in ('Vipul','Satish');

The screenshot shows the Oracle Application Express interface. In the top navigation bar, there are tabs for Home, Application Builder, SQL Workshop (selected), Team Development, and Administration. Below the navigation bar, the URL is 127.0.0.1:8080/apex/f?p=4500:1003:3186514920682929::NO:: and the page title is ORACLE Application Express.

In the main content area, there is a SQL editor window with the following SQL code:

```

Select REPLACE(FIRST_NAME,'a','A') FROM WORKER;
Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' FROM WORKER;
Select * FROM WORKER ORDER BY FIRST_NAME ASC;
Select * FROM WORKER ORDER BY FIRST_NAME ASC, DEPARTMENT DESC;
Select * FROM WORKER WHERE FIRST_NAME IN ('Vipul','Satish');
Select * FROM WORKER WHERE FIRST_NAME NOT IN ('Vipul','Satish');
Select * FROM WORKER WHERE DEPARTMENT LIKE 'Admin%';
Select * FROM WORKER WHERE FIRST_NAME LIKE '%an%';
Select * FROM WORKER WHERE FIRST_NAME LIKE 'Ka%';

```

Below the SQL editor, there is a results grid showing six rows of data:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	02/20/0014	HR
2	Niharika	Verma	80000	06/11/0014	Admin
3	Vishal	Singhal	300000	02/20/0014	HR
5	Vivek	Bhati	500000	06/11/0014	Admin
8	Geetika	Chauhan	90000	04/11/0014	Admin
4	Amitabh	Singh	500000	02/20/0014	Admin

Text at the bottom of the results grid: 6 rows returned in 0.07 seconds [Download](#)

Bottom right corner: Application Express 4.0.2.00.09 Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

15. Write An SQL Query To Print Details Of Workers With DEPARTMENT Name As “Admin”.

- Select * from Worker where DEPARTMENT like 'Admin%';

The screenshot shows the Oracle Application Express interface. In the SQL Workshop, a query is run:

```

Select REPLACE(FIRST_NAME,'a','A') from Worker;
Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' from Worker;
Select * from Worker order by FIRST_NAME asc;
Select * from Worker order by FIRST_NAME desc,DEPARTMENT desc;
Select * from Worker where FIRST_NAME in ('Vipul','Satish');
Select * from Worker where FIRST_NAME not in ('Vipul','Satish');
Select * from Worker where DEPARTMENT like 'Admin%';
Select * from Worker where FIRST_NAME like 'Na%';
Select * from Worker where FIRST_NAME like 'Na';

```

The results show four rows returned in 0.04 seconds:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
2	Niharika	Verma	80000	06/11/2014	Admin
5	Vivek	Bhati	500000	06/11/2014	Admin
8	Geetika	Chauhan	90000	04/11/2014	Admin
4	Amitabh	Singh	500000	02/20/2014	Admin

4 rows returned in 0.04 seconds [Download](#)

Application Express 4.0.2.0.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

16. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Contains „A“.

- Select * from Worker where FIRST_NAME like '%a%';

The screenshot shows the Oracle Application Express interface. In the SQL Workshop, a query is run:

```

Select REPLACE(FIRST_NAME,'a','A') from Worker;
Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' from Worker;
Select * from Worker order by FIRST_NAME asc;
Select * from Worker order by FIRST_NAME desc,DEPARTMENT desc;
Select * from Worker where FIRST_NAME in ('Vipul','Satish');
Select * from Worker where FIRST_NAME not in ('Vipul','Satish');
Select * from Worker where DEPARTMENT like 'Admin%';
Select * from Worker where FIRST_NAME like 'Na%';
Select * from Worker where FIRST_NAME like 'Na';

```

The results show six rows returned in 0.05 seconds:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	02/20/2014	HR
2	Niharika	Verma	80000	06/11/2014	Admin
3	Vishal	Singhal	300000	02/20/2014	HR
8	Geetika	Chauhan	90000	04/11/2014	Admin
4	Amitabh	Singh	500000	02/20/2014	Admin
7	Satish	Kumar	75000	01/20/2014	Account

6 rows returned in 0.05 seconds [Download](#)

Application Express 4.0.2.0.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

17. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Ends With „A“.

- Select * from Worker where FIRST_NAME like '%a';

The screenshot shows the Oracle Application Express interface. In the SQL Workshop tab, the following SQL code is executed:

```

Select * from Worker where FIRST_NAME not in ('Vipul','Satish');
Select * from Worker where DEPARTMENT like 'Admin%';
Select * from Worker where FIRST_NAME like 'Na%';
Select * from Worker where FIRST_NAME like 'Na$';
Select * from Worker where FIRST_NAME like '_____h';
Select * from Worker where SALARY between 100000 and 500000;
Select * from Worker where year(JOINING_DATE) = 2014 and month(JOINING_DATE) = 2;
SELECT COUNT(*) FROM worker WHERE DEPARTMENT = 'Admin';

```

The results show three rows of data:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	02/20/2014	HR
2	Niharika	Verma	80000	06/11/2014	Admin
8	Geetika	Chauhan	90000	04/11/2014	Admin

3 rows returned in 0.05 seconds

18. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Ends With „H“ And Contains Six Alphabets.

- Select * from Worker where FIRST_NAME like '_____h';

The screenshot shows the Oracle Application Express interface. In the SQL Workshop tab, the following SQL code is executed:

```

Select * from Worker where FIRST_NAME not in ('Vipul','Satish');
Select * from Worker where DEPARTMENT like 'Admin%';
Select * from Worker where FIRST_NAME like 'Na%';
Select * from Worker where FIRST_NAME like 'Na$';
Select * from Worker where FIRST_NAME like '_____h';
Select * from Worker where SALARY between 100000 and 500000;
Select * from Worker where year(JOINING_DATE) = 2014 and month(JOINING_DATE) = 2;
SELECT COUNT(*) FROM worker WHERE DEPARTMENT = 'Admin';

SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS Worker_Name, DEPARTMENT

```

The results show one row of data:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
7	Satish	Kumar	75000	01/20/2014	Account

1 rows returned in 0.00 seconds

19. Write An SQL Query To Print Details Of The Workers Whose SALARY Lies Between 100000 And 500000

- Select * from Worker where SALARY between 100000 and 500000;

```

Select * from Worker where FIRST_NAME not in ('Vipul','Satish');
Select * from Worker where DEPARTMENT like 'Admin%';
Select * from Worker where FIRST_NAME like '%S%';
Select * from Worker where FIRST_NAME like 'Na%';
Select * from Worker where FIRST_NAME like '____h';
Select * from Worker where SALARY between 100000 and 500000];
Select * from Worker where year(JOINING_DATE) = 2014 and month(JOINING_DATE) = 2;
SELECT COUNT(*) FROM worker WHERE DEPARTMENT = 'Admin';

```

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	02/20/0014	HR
3	Vishal	Singhal	300000	02/20/0014	HR
5	Vivek	Bhati	500000	06/11/0014	Admin
4	Amitabh	Singh	500000	02/20/0014	Admin
6	Vipul	Diwan	200000	06/11/0014	Account

5 rows returned in 0.03 seconds [Download](#)

Workspace: CSE User: CSE Application Express 4.0.2.0.09 Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

20. Write An SQL Query To Print Details Of The Workers Who Have Joined In Feb“2014.

- select * from Worker where extract(Year from Joining_Date)=2014 and extract(Month from Joining_Date)=02;

```

HAVING COUNT(*) > 2;
select * from Worker where extract(Year from Joining_Date)>2014 and extract(Month from Joining_Date)>02;

```

no data found

Workspace: CSE User: CSE Application Express 4.0.2.0.09 Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

21. Write An SQL Query To Fetch The Count Of Employees Working In The Department „Admin“

- SELECT COUNT(*) FROM worker WHERE DEPARTMENT = 'Admin';

The screenshot shows the Oracle Application Express interface. In the top navigation bar, the URL is 127.0.0.1:8080/apex/f?p=4500:1003:318651492068299::NO:: and the page title is 'SQL Commands'. The main content area contains the following SQL code:

```

Select * from Worker where FIRST_NAME not in ('Vipul','Satish');
Select * from Worker where DEPARTMENT like 'Admin%';
Select * from Worker where FIRST_NAME like 'Na%';
Select * from Worker where FIRST_NAME like 'Na';
Select * from Worker where FIRST_NAME like '____h';
Select * from Worker where SALARY between 100000 and 500000;
Select * from Worker where year(JOINING_DATE) = 2014 and month(JOINING_DATE) = 2;
SELECT COUNT(*) FROM worker WHERE DEPARTMENT = 'Admin';

SELECT CONCAT(FIRST_NAME, LAST_NAME) AS WORKER_NAME, SALARY

```

The results pane shows a single row of data:

COUNT()
4

Below the results, it says '1 rows returned in 0.01 seconds'.

22. Write An SQL Query To Fetch Worker Names With Salaries ≥ 50000 And ≤ 100000 .

- SELECT CONCAT(FIRST_NAME,LAST_NAME) AS WORKER_NAME,SALARY FROM WORKER WHERE SALARY BETWEEN 50000 AND 100000;

The screenshot shows the Oracle Application Express interface. In the top navigation bar, the URL is 127.0.0.1:8080/apex/f?p=4500:1003:2571053880063041::NO:: and the page title is 'SQL Commands'. The main content area contains the following SQL code:

```

SELECT CONCAT(FIRST_NAME, LAST_NAME) AS WORKER_NAME, SALARY FROM WORKER WHERE SALARY BETWEEN 50000 AND 100000;

```

The results pane shows four rows of data:

WORKER_NAME	SALARY
MonikaArora	100000
NiharikaVerma	80000
GeetikaChauhan	90000
SatishKumar	75000

Below the results, it says '4 rows returned in 0.36 seconds'.

23. Write An SQL Query To Fetch The No. Of Workers for Each Department In The Descending Order.

- SELECT DEPARTMENT, count(WORKER_ID) No_of_Workers
FROM worker
GROUP BY DEPARTMENT
ORDER BY No_of_Workers DESC;

The screenshot shows the Oracle Application Express interface. In the top navigation bar, 'SQL Commands' is selected. The main area contains the following SQL code:

```

SELECT COUNT(*) FROM worker WHERE DEPARTMENT = 'Admin';

SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) As Worker_Name, Salary
FROM worker
WHERE WORKER_ID IN
(SELECT WORKER_ID FROM worker
WHERE SALARY BETWEEN 50000 AND 100000);

SELECT DEPARTMENT, count(WORKER_ID) No_of_Workers
FROM worker
GROUP BY DEPARTMENT
ORDER BY No_of_Workers DESC;

SELECT DISTINCT W.FIRST_NAME, T.WORKER_TITLE

```

Below the code, there is a results table:

DEPARTMENT	NO_OF_WORKERS
Admin	4
HR	2
Account	2

At the bottom, it says "3 rows returned in 0.01 seconds".

24. Write An SQL Query To Print Details Of The Workers Who Are Also Managers.

➤ SELECT DISTINCT W.FIRST_NAME, T.WORKER_TITLE
 FROM Worker W
 INNER JOIN Title T
 ON W.WORKER_ID = T.WORKER_REF_ID
 AND T.WORKER_TITLE in ('Manager');

The screenshot shows the Oracle Application Express interface. In the top navigation bar, 'SQL Commands' is selected. The main area contains the following SQL code:

```

WHERE WORKER_ID IN
(SELECT WORKER_ID FROM worker
WHERE SALARY BETWEEN 50000 AND 100000);

SELECT DEPARTMENT, count(WORKER_ID) No_of_Workers
FROM worker
GROUP BY DEPARTMENT
ORDER BY No_of_Workers DESC;

SELECT DISTINCT W.FIRST_NAME, T.WORKER_TITLE
FROM Worker W
INNER JOIN Title T
ON W.WORKER_ID = T.WORKER_REF_ID
AND T.WORKER_TITLE IN ('Manager')

```

Below the code, there is a results table:

FIRST_NAME	WORKER_TITLE
Monika	Manager
Vivek	Manager

At the bottom, it says "2 rows returned in 0.19 seconds".

25. Write An SQL Query To Fetch Duplicate Records Having Matching Data In Some Fields Of A Table.

➤ SELECT WORKER_TITLE, COUNT(*)
 FROM Title
 GROUP BY WORKER_TITLE
 HAVING COUNT(*) > 1;

```

    INNER JOIN Title t ON M.WORKER_REF_ID = T.WORKER_REF_ID
    AND T.WORKER_TITLE IN ('Manager');

    SELECT WORKER_TITLE, AFFECTED_FROM, COUNT(*)
    FROM Title
    GROUP BY WORKER_TITLE, AFFECTED_FROM
    HAVING COUNT(*) > 1;
  
```

Results

WORKER_TITLE	AFFECTED_FROM	COUNT()
Executive	06/11/0016	3
Lead	06/11/0016	2

2 rows returned in 0.08 seconds [Download](#)

Application Express 4.0.2.0.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

26. Write An SQL Query To Show Only Odd Rows From worker Table.

➤ `SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) <> 0;`

```

    SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) <> 0;
    SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) = 0;
    CREATE TABLE WorkerClone LIKE Worker;
    (SELECT * FROM Worker)
    INTERSECT
    (SELECT * FROM WorkerClone);
    SELECT * FROM Worker
    MINUS
    SELECT * FROM Title;
    SELECT SYSDATE FROM DUAL;
  
```

Results

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	02/20/0014	HR
3	Vishal	Singhal	300000	02/20/0014	HR
5	Vivek	Bhati	500000	06/11/0014	Admin
7	Salish	Kumar	75000	01/20/0014	Account

4 rows returned in 0.00 seconds [Download](#)

Application Express 4.0.2.0.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

27. Write An SQL Query To Show Only Even Rows From worker Table.

➤ `SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) = 0;`

The screenshot shows the Oracle Application Express interface. In the top navigation bar, 'SQL Commands' is selected. The main area contains the following SQL code:

```

SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) <> 0;
SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) = 0;
CREATE TABLE WorkerClone LIKE Worker;
(SELECT * FROM Worker)
INTERSECT
(SELECT * FROM WorkerClone);
SELECT * FROM Worker
MINUS
SELECT * FROM Title;

SELECT SYSDATE FROM DUAL;

```

Below the code, the results are displayed in a table:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
2	Niharika	Verma	80000	06/11/0014	Admin
8	Geetika	Chauhan	90000	04/11/0014	Admin
4	Amitabh	Singh	500000	02/20/0014	Admin
6	Vigul	Diwan	200000	06/11/0014	Account

4 rows returned in 0.00 seconds. A 'Download' link is present.

Bottom status bar: Application Express 4.0.2.00.09, Workspace: CSE User: CSE, Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

28. Write An SQL Query To Clone New_Worker Table From Worker Table.

- create table WORKER_CLONE as select * from WORKER

The screenshot shows the Oracle Application Express interface. In the top navigation bar, 'SQL Commands' is selected. The main area contains the following SQL code:

```

create table WORKER_CLONE as select * from WORKER
DESC WORKER_CLONE;

```

Below the code, the results are displayed in a table:

Object Type TABLE Object WORKER_CLONE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WORKER_CLONE	WORKER_ID	NUMBER	-	5	0	-	✓	-	-
	FIRST_NAME	VARCHAR2	30	-	-	-	✓	-	-
	LAST_NAME	VARCHAR2	30	-	-	-	✓	-	-
	SALARY	NUMBER	-	7	0	-	✓	-	-
	JOINING_DATE	DATE	7	-	-	-	✓	-	-
	DEPARTMENT	VARCHAR2	7	-	-	-	✓	-	-

1 - 6

Bottom status bar: Application Express 4.0.2.00.09, Workspace: CSE User: CSE, Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

29. Write An SQL Query To Fetch Intersecting Records Of Worker and New_Worker Tables.

- insert into WORKER_CLONE select * from WORKER;

Inbox - piyushchandrachandra@ 127.0.0.1:8080/apex/r?p=4500:1003:2571053880063041:NO

Home Application Builder SQL Workshop Team Development Administration

Autocommit Rows 10 Save Run

```
create table WORKER_CLONE as select * from WORKER
DESC WORKER_CLONE;
insert into WORKER_CLONE select * from WORKER;
SELECT * FROM WORKER_CLONE;
```

Results Explain Describe Saved SQL History

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	02/20/0014	HR
2	Niharika	Verma	80000	06/11/0014	Admin
3	Vishal	Singhal	300000	02/20/0014	HR
5	Vivek	Bhati	500000	06/11/0014	Admin
8	Geetika	Chauhan	90000	04/11/0014	Admin
4	Amitabh	Singh	500000	02/20/0014	Admin
6	Vipul	Diwan	200000	05/11/0014	Account
7	Satish	Kumar	75000	01/20/0014	Account
1	Monika	Arora	100000	02/20/0014	HR
2	Niharika	Verma	80000	06/11/0014	Admin

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.16 seconds Download

30. Write An SQL Query To Show Records From Worker table That Title Table Does Not Have.

- `SELECT WORKER_ID , JOINING_DATE FROM WORKER MINUS SELECT WORKER_REF_ID , AFFECTED_FROM FROM TITLE;`

Inbox - piyushchandrachandra@ 127.0.0.1:8080/apex/r?p=4500:1003:2571053880063041:NO

Home Application Builder SQL Workshop Team Development Administration

Autocommit Rows 10 Save Run

```
create table WORKER_CLONE as select * from WORKER
DESC WORKER_CLONE;
insert into WORKER_CLONE select * from WORKER;
SELECT * FROM WORKER_CLONE;

SELECT WORKER_ID , JOINING_DATE FROM WORKER MINUS SELECT WORKER_REF_ID , AFFECTED_FROM FROM TITLE;
```

Results Explain Describe Saved SQL History

WORKER_ID	JOINING_DATE
1	02/20/0014
2	06/11/0014
3	02/20/0014
4	02/20/0014
5	06/11/0014
6	06/11/0014
7	01/20/0014
8	04/11/0014

8 rows returned in 0.27 seconds Download

Application Express 4.0.2.00.09
Workspace: CSE User: CSE Language: en Copyright © 1999, 2010, Oracle. All rights reserved.

31. Write An SQL Query To Show The Current Date.

- `SELECT SYSDATE FROM DUAL;`

The screenshot shows the Oracle Application Express interface. In the top navigation bar, 'SQL Commands' is selected. The main area contains the following SQL code:

```

SELECT * INTO WorkerClone FROM WORKER WHERE 1 = 0;
(SELECT * FROM Worker)
INTERSECT
(SELECT * FROM WorkerClone);
SELECT * FROM Worker
MINUS
SELECT * FROM Title;

SELECT SYSDATE FROM DUAL;]

SELECT * FROM (SELECT * FROM Worker ORDER BY Salary DESC)
WHERE ROWNUM <= 10;

SELECT TOP 1 Salary
FROM (

```

Below the code, the results section shows a single row for 'SYSDATE':

SYSDATE
12/02/2020

Text at the bottom indicates '1 rows returned in 0.13 seconds'.

32. Write An SQL Query To Show The Top N (Say 5) Records Of Worker Table.

- **SELECT * FROM (SELECT * FROM Worker ORDER BY Salary DESC)
WHERE ROWNUM <= 10;**

The screenshot shows the Oracle Application Express interface. In the top navigation bar, 'SQL Commands' is selected. The main area contains the following SQL code:

```

SELECT * INTO WorkerClone FROM WORKER WHERE 1 = 0;
(SELECT * FROM Worker)
INTERSECT
(SELECT * FROM WorkerClone);
SELECT * FROM Worker
MINUS
SELECT * FROM Title;

SELECT SYSDATE FROM DUAL;]

SELECT * FROM (SELECT * FROM Worker ORDER BY Salary DESC)
WHERE ROWNUM <= 10;

SELECT TOP 1 Salary
FROM (

```

Below the code, the results section shows a table with 8 rows of worker data:

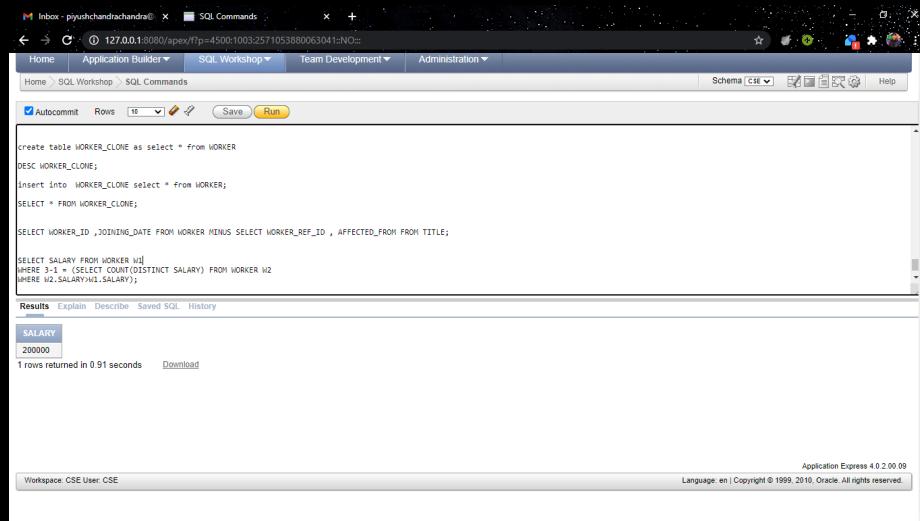
WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
5	Vivek	Bhati	500000	06/11/0014	Admin
4	Amitabh	Singh	500000	02/20/0014	Admin
3	Vishal	Singhal	300000	02/20/0014	HR
6	Vipul	Diwan	200000	06/11/0014	Account
1	Monika	Arora	100000	02/20/0014	HR
8	Geetika	Chauhan	90000	04/11/0014	Admin
2	Niharika	Verma	80000	06/11/0014	Admin
7	Satish	Kumar	75000	01/20/0014	Account

Text at the bottom indicates '8 rows returned in 0.04 seconds'.

33. Write An SQL Query To Determine The Nth (Say 3) Highest Salary From Worker Table.

- **SELECT SALARY FROM WORKER W1**

WHERE 3~1 = (SELECT COUNT(DISTINCT SALARY) FROM WORKER W2
 WHERE W2.SALARY>W1.SALARY);



```

create table WORKER_CLONE as select * from WORKER
DESC WORKER_CLONE;
insert into WORKER_CLONE select * from WORKER;
SELECT * FROM WORKER_CLONE;
SELECT WORKER_ID ,JOINING_DATE FROM WORKER MINUS SELECT WORKER_REF_ID , AFFECTED_FROM FROM TTITLE;
SELECT SALARY FROM WORKER W1
WHERE 3~1 = (SELECT COUNT(DISTINCT SALARY) FROM WORKER W2
WHERE W2.SALARY>W1.SALARY);

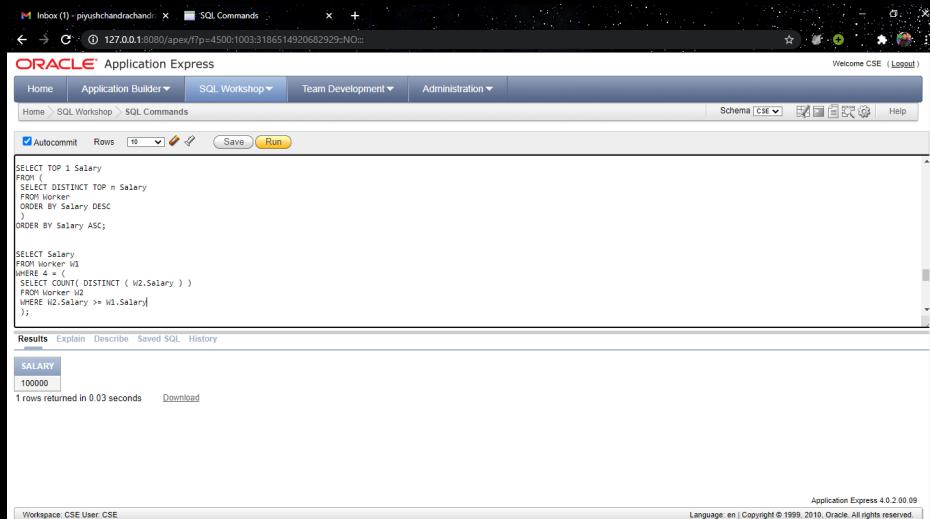
Results Explain Describe Saved SQL History
SALARY
200000
1 rows returned in 0.91 seconds Download

```

Application Express 4.0.2.0.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

34. Write An SQL Query To Determine The 5th Highest Salary Without Using TOP Or Limit Method from Worker Table.

➤ SELECT Salary
 FROM Worker W1
 WHERE 4 = (
 SELECT COUNT(DISTINCT (W2.Salary))
 FROM Worker W2
 WHERE W2.Salary >= W1.Salary
);



```

SELECT TOP 1 Salary
FROM (
SELECT DISTINCT TOP n Salary
FROM Worker
ORDER BY Salary DESC
)
ORDER BY Salary ASC;

SELECT Salary
FROM WORKER W1
WHERE 4 = (
  SELECT COUNT( DISTINCT ( W2.Salary ) )
  FROM Worker W2
  WHERE W2.Salary >= W1.Salary
);

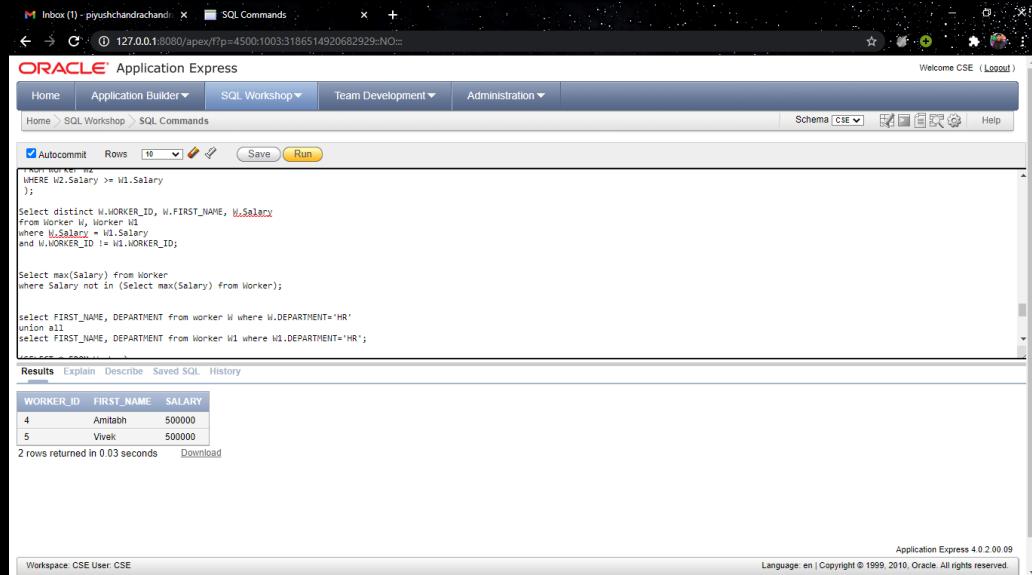
```

Welcome CSE (Logged in)
Application Express 4.0.2.0.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

35. Write An SQL Query To Fetch The List Of Employees With The Same Salary from Worker Table.

➤ Select distinct W.WORKER_ID, W.FIRST_NAME, W.Salary

from Worker W, Worker W1
 where W.Salary = W1.Salary
 and W.WORKER_ID != W1.WORKER_ID;



```

 1< Oracle Application Express
 2< Home Application Builder SQL Workshop Team Development Administration
 3< Home SQL Workshop SQL Commands
 4< Schema CSE Help
 5< Autocommit Rows 10 Save Run
 6< NOT WORKER.W2
 7< WHERE W2.Salary >= W1.Salary
 8<
 9< Select distinct W.WORKER_ID, W.FIRST_NAME, W.Salary
10< From Worker W, Worker W1
11< Where W.Salary = W1.Salary
12< And W.WORKER_ID != W1.WORKER_ID;
13<
14< Select max(Salary) from Worker
15< Where Salary not in (Select max(Salary) from Worker);
16<
17< select FIRST_NAME, DEPARTMENT from worker W where W.DEPARTMENT='HR'
18< union all
19< select FIRST_NAME, DEPARTMENT from Worker W1 where W1.DEPARTMENT='HR';
  
```

Results

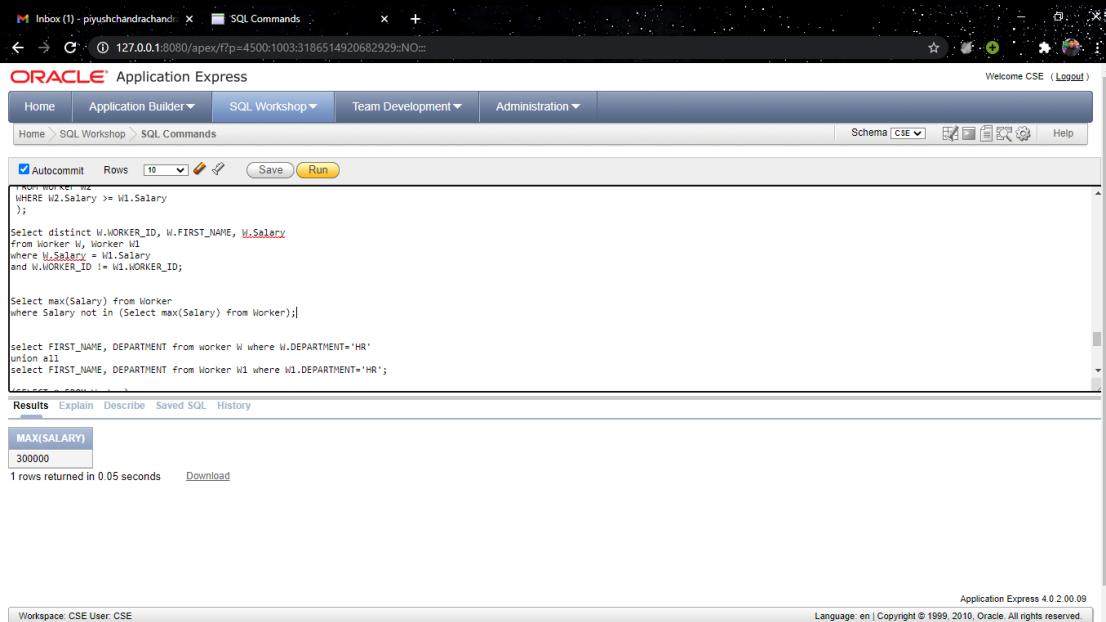
WORKER_ID	FIRST_NAME	SALARY
4	Amitabh	500000
5	Vivek	500000

2 rows returned in 0.03 seconds [Download](#)

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

36. Write An SQL Query To Show The Second Highest Salary From Worker Table.

- Select max(Salary) from Worker
 where Salary not in (Select max(Salary) from Worker);



```

 1< Oracle Application Express
 2< Home Application Builder SQL Workshop Team Development Administration
 3< Home SQL Workshop SQL Commands
 4< Schema CSE Help
 5< Autocommit Rows 10 Save Run
 6< NOT WORKER.W2
 7< WHERE W2.Salary >= W1.Salary
 8<
 9< Select distinct W.WORKER_ID, W.FIRST_NAME, W.Salary
10< From Worker W, Worker W1
11< Where W.Salary = W1.Salary
12< And W.WORKER_ID != W1.WORKER_ID;
13<
14< Select max(Salary) from Worker
15< Where Salary not in (Select max(Salary) from Worker);
16<
17< select FIRST_NAME, DEPARTMENT from worker W where W.DEPARTMENT='HR'
18< union all
19< select FIRST_NAME, DEPARTMENT from Worker W1 where W1.DEPARTMENT='HR';
  
```

Results

MAX(SALARY)
300000

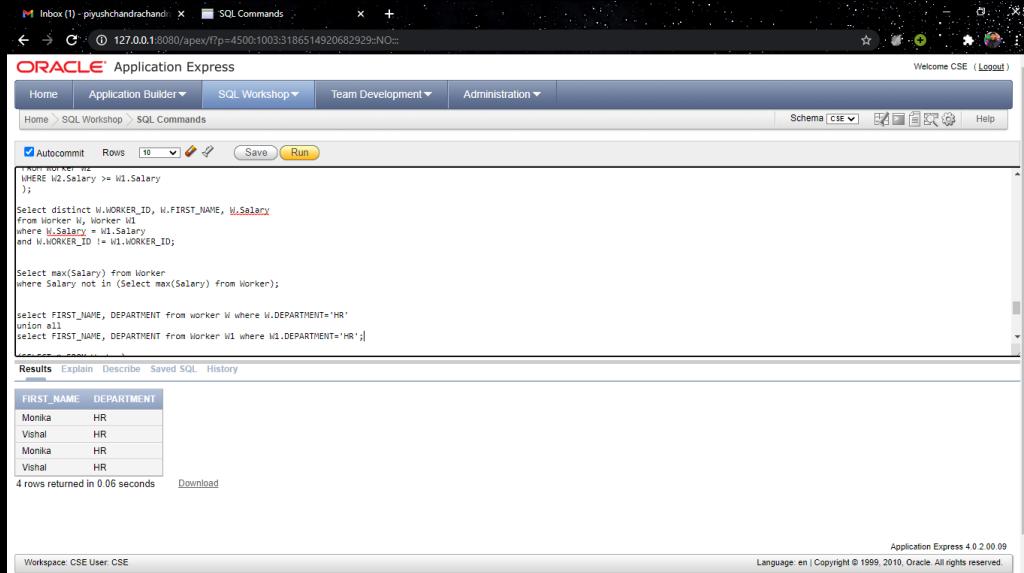
1 rows returned in 0.05 seconds [Download](#)

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

37. Write An SQL Query To Show One Row Twice In Results From Worker Table.

- select FIRST_NAME, DEPARTMENT from worker W where W.DEPARTMENT='HR'
 union all

select FIRST_NAME, DEPARTMENT from Worker W1 where W1.DEPARTMENT='HR';



The screenshot shows the Oracle Application Express interface. The SQL Commands page displays the following SQL code:

```
SELECT FIRST_NAME, DEPARTMENT
FROM Worker W1
WHERE W2.Salary >= W1.Salary
;

Select distinct W.WORKER_ID, W.FIRST_NAME, W.Salary
from Worker W, Worker W1
where W.Salary = W1.Salary
and W.WORKER_ID != W1.WORKER_ID;

Select max(Salary) from Worker
where Salary not in (Select max(Salary) from Worker);

select FIRST_NAME, DEPARTMENT from worker W where W.DEPARTMENT='HR'
union all
select FIRST_NAME, DEPARTMENT from Worker W1 where W1.DEPARTMENT='HR';
```

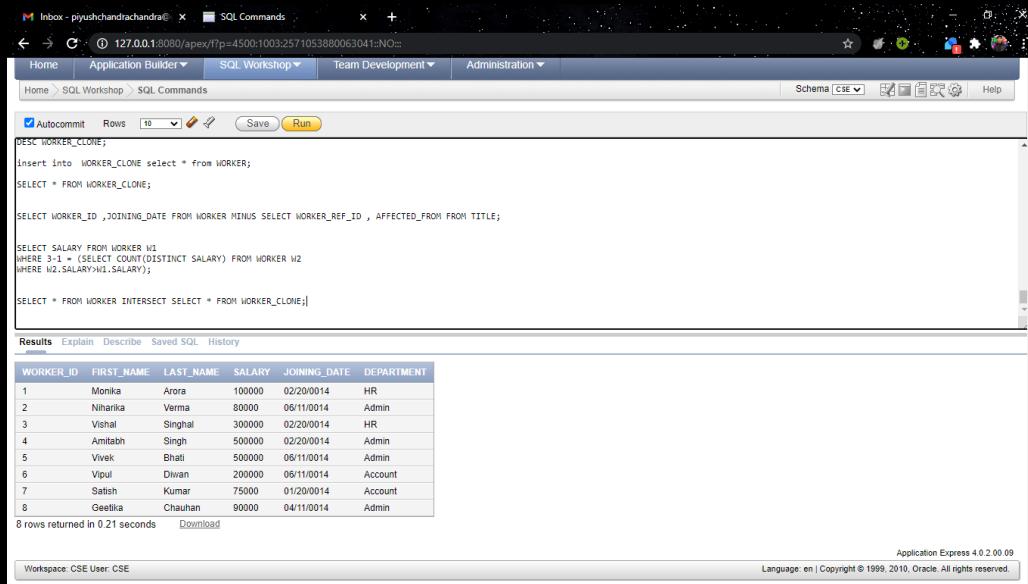
The Results section shows the output:

FIRST_NAME	DEPARTMENT
Monika	HR
Vishal	HR
Monika	HR
Vishal	HR

4 rows returned in 0.06 seconds

38. Write An SQL Query To Fetch Intersecting Records Of Worker and New_Worker Tables.

- SELECT * FROM WORKER INTERSECT SELECT * FROM WORKER_CLONE;



The screenshot shows the Oracle Application Express interface. The SQL Commands page displays the following SQL code:

```
DESC WORKER_CLONE;
insert into WORKER_CLONE select * from WORKER;
SELECT * FROM WORKER_CLONE;

SELECT WORKER_ID ,JOINING_DATE FROM WORKER MINUS SELECT WORKER_REF_ID , AFFECTED_FROM FROM TITLE;

SELECT SALARY FROM WORKER W1
WHERE 3-1 = (SELECT COUNT(DISTINCT SALARY) FROM WORKER W2
WHERE W2.SALARY>W1.SALARY);

SELECT * FROM WORKER INTERSECT SELECT * FROM WORKER_CLONE;
```

The Results section shows the output:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	02/20/0014	HR
2	Niharika	Verma	80000	06/11/0014	Admin
3	Vishal	Singhal	300000	02/20/0014	HR
4	Amitabh	Singh	500000	02/20/0014	Admin
5	Vivek	Bhati	500000	06/11/0014	Admin
6	Vipul	Diwan	200000	06/11/0014	Account
7	Satish	Kumar	75000	01/20/0014	Account
8	Geetika	Chauhan	90000	04/11/0014	Admin

8 rows returned in 0.21 seconds

39. Write An SQL Query To Fetch The First 50% Records From Worker Table.

- SELECT *
FROM WORKER

WHERE WORKER_ID <= (SELECT COUNT(WORKER_ID)/2 FROM WORKER);

```

SELECT * FROM WORKER_CLONE;

SELECT WORKER_ID ,JOINING_DATE FROM WORKER MINUS SELECT WORKER_REF_ID , AFFECTED_FROM FROM TITLE;

SELECT SALARY FROM WORKER W1
WHERE 3-1 = (SELECT COUNT(DISTINCT SALARY) FROM WORKER W2
WHERE W2.SALARY>W1.SALARY);

SELECT * FROM WORKER INTERSECT SELECT * FROM WORKER_CLONE;

SELECT *
FROM WORKER
WHERE WORKER_ID <= (SELECT COUNT(WORKER_ID)/2 FROM WORKER);

```

Results

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	02/20/0014	HR
2	Niharika	Vermi	80000	06/11/0014	Admin
3	Vishal	Singhal	300000	02/20/0014	HR
4	Amitabh	Singh	500000	02/20/0014	Admin

4 rows returned in 0.07 seconds [Download](#)

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

40. Write An SQL Query To Fetch The Departments That Have Less Than Five People In It from Worker Table.

- **SELECT DEPARTMENT, COUNT(WORKER_ID) FROM WORKER GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;**

```

SELECT WORKER_ID ,JOINING_DATE FROM WORKER MINUS SELECT WORKER_REF_ID , AFFECTED_FROM FROM TITLE;

SELECT SALARY FROM WORKER W1
WHERE 3-1 = (SELECT COUNT(DISTINCT SALARY) FROM WORKER W2
WHERE W2.SALARY>W1.SALARY);

SELECT * FROM WORKER INTERSECT SELECT * FROM WORKER_CLONE;

SELECT *
FROM WORKER
WHERE WORKER_ID <= (SELECT COUNT(WORKER_ID)/2 FROM WORKER);

SELECT DEPARTMENT , COUNT(WORKER_ID) FROM WORKER GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;

```

Results

DEPARTMENT	COUNT(WORKER_ID)
HR	2
Account	2
Admin	4

3 rows returned in 0.14 seconds [Download](#)

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

41. Write An SQL Query To Show All Departments Along With The Number Of People In There from Worker Table.

- SELECT DEPARTMENT, COUNT (DEPARTMENT) AS “NUMBER OF WORKERS” FROM WORKER GROUP BY DEPARTMENT;

```

Select INSTR(FIRST_NAME,'A') from WORKER where FIRST_NAME = 'Amitabh';

Select CONCAT(CONCAT(FIRST_NAME, ' '),LAST_NAME)AS "COMPLETE_NAME" FROM WORKER;

```

COMPLETE_NAME
Monika Arora
Niharika Verma
Vishal Singhal
Vivek Bhati
Geetika Chauhan
Amitabh Singh
Vipul Diwan
Satish Kumar

8 rows returned in 0.02 seconds Download

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

42. Write An SQL Query To Show The Last Record From Worker Table.

- Select * from Worker where WORKER_ID = (SELECT max(WORKER_ID) from Worker);

```

SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary >= b.Salary) order by a.Salary desc;

SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;

SELECT t1.DEPARTMENT,t1.FIRST_NAME,t1.Salary FROM(SELECT max(Salary) as TotalSalary,DEPARTMENT FROM Worker group by DEPARTMENT) as TempNew
Inner Join Worker t1 on TempNew.DEPARTMENT=t1.DEPARTMENT
and TempNew.TotalSalary=t1.Salary;

SELECT * FROM Worker WHERE WORKER_ID <=5
UNION
SELECT * FROM (SELECT * FROM Worker W order by W.WORKER_ID DESC) AS W1 WHERE W1.WORKER_ID <=5;

Select * from Worker where WORKER_ID = (SELECT min(WORKER_ID) from Worker);
Select * from Worker where WORKER_ID = (SELECT max(WORKER_ID) from Worker);


```

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
8	Geetika	Chauhan	90000	04/11/2014	Admin

1 rows returned in 0.08 seconds Download

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

43. Write An SQL Query To Fetch The First Row Of Worker Table.

- Select * from Worker where WORKER_ID = (SELECT min(WORKER_ID) from Worker);

The screenshot shows the Oracle Application Express interface with a SQL Workshop tab selected. The query is as follows:

```

SELECT distinct Salary from worker a WHERE N >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;
SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary >= b.Salary) order by a.Salary desc;
SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;
SELECT t.DEPARTMENT,t.FIRST_NAME,t.Salary from(SELECT max(Salary) as TotalSalary,DEPARTMENT from Worker group by DEPARTMENT) as TempNew
Inner Join Worker t on TempNew.DEPARTMENT=t.DEPARTMENT
and TempNew.TotalSalary=t.Salary;
SELECT * FROM Worker WHERE WORKER_ID <=5
UNION
SELECT * FROM (SELECT * FROM Worker W order by W.WORKER_ID DESC) AS W1 WHERE W1.WORKER_ID <=5;
Select * from Worker where WORKER_ID = (SELECT min(WORKER_ID) from Worker);

```

The results table shows one row:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	02/20/0014	HR

1 rows returned in 0.05 seconds [Download](#)

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

44. Write An SQL Query To Fetch The Last Five Records From Worker Table.

➤ `SELECT * FROM(SELECT * FROM WORKER ORDER BY WORKER_ID DESC) REV_WORKER
WHERE ROWNUM <= 5
ORDER BY ROWNUM DESC;`

The screenshot shows the Oracle Application Express interface with a SQL Workshop tab selected. The query is as follows:

```

SELECT *
FROM WORKER
WHERE WORKER_ID <= (SELECT COUNT(WORKER_ID)/2 FROM WORKER);

SELECT DEPARTMENT, COUNT(WORKER_ID) FROM WORKER GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;

SELECT DEPARTMENT, COUNT(DEPARTMENT) AS 'NUMBER OF WORKERS' FROM WORKER GROUP BY DEPARTMENT;

SELECT DISTINCT SALARY FROM WORKER a WHERE 1 >= (SELECT COUNT(DISTINCT SALARY) FROM WORKER b WHERE a.SALARY <= b.SALARY) ORDER BY a.SALARY DESC;

SELECT T.DEPARTMENT, T.FIRST_NAME, T.SALARY FROM(SELECT MAX(SALARY) AS "TOTAL_SALARY",DEPARTMENT FROM WORKER GROUP BY DEPARTMENT) TEMPNEW
INNER JOIN WORKER T ON TEMPNEW.DEPARTMENT=T.DEPARTMENT AND TEMPNEW.TOTAL_SALARY=T.SALARY;

SELECT * FROM(SELECT * FROM WORKER ORDER BY WORKER_ID DESC) REV_WORKER
WHERE ROWNUM <= 5
ORDER BY ROWNUM DESC;

```

The results table shows five rows:

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
4	Amitabh	Singh	500000	02/20/0014	Admin
5	Vivek	Bhati	500000	06/11/0014	Admin
6	Vipul	Diwan	200000	06/11/0014	Account
7	Satish	Kumar	75000	01/20/0014	Account

5 rows returned in 0.10 seconds [Download](#)

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

45. Write An SQL Query To Print The Name Of Employees Having The Highest Salary In Each Department from Worker Table.

- SELECT T.DEPARTMENT, T.FIRST_NAME, T.SALARY FROM(SELECT MAX(SALARY) AS "TOTAL_SALARY", "DEPARTMENT" FROM WORKER GROUP BY DEPARTMENT) TEMPNEW
INNER JOIN WORKER T ON TEMPNEW.DEPARTMENT=T.DEPARTMENT AND
TEMPNEW.TOTAL_SALARY=T.SALARY;

```

SELECT * FROM WORKER INTERSECT SELECT * FROM WORKER_CLONE;
SELECT *
FROM WORKER
WHERE WORKER_ID <= (SELECT COUNT(WORKER_ID)/2 FROM WORKER);

SELECT DEPARTMENT, COUNT(WORKER_ID) FROM WORKER GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;

SELECT DEPARTMENT, COUNT(DEPARTMENT) AS 'NUMBER OF WORKERS' FROM WORKER GROUP BY DEPARTMENT;

SELECT DISTINCT SALARY FROM WORKER a WHERE 1<=(SELECT COUNT(DISTINCT SALARY) FROM WORKER b WHERE a.SALARY <= b.SALARY) ORDER BY a.SALARY DESC;
SELECT T.DEPARTMENT, T.FIRST_NAME, T.SALARY FROM(SELECT MAX(SALARY) AS "TOTAL_SALARY", "DEPARTMENT" FROM WORKER GROUP BY DEPARTMENT) TEMPNEW
INNER JOIN WORKER T ON TEMPNEW.DEPARTMENT=T.DEPARTMENT AND TEMPNEW.TOTAL_SALARY=T.SALARY;
    
```

Results

DEPARTMENT	FIRST_NAME	SALARY
HR	Vishal	300000
Admin	Vivek	500000
Admin	Amitabh	500000
Account	Vipul	200000

4 rows returned in 0.31 seconds [Download](#)

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

46. Write An SQL Query To Fetch Three Max Salaries From Worker Table.

- SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;

```

SELECT DEPARTMENT, COUNT(WORKER_ID) as 'Number of Workers' FROM WORKER GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;

SELECT FIRST_NAME, SALARY from WORKER WHERE SALARY=(SELECT max(SALARY) from WORKER);

SELECT DEPARTMENT, sum(Salary) from WORKER group by DEPARTMENT;

SELECT distinct Salary from WORKER a WHERE N >= (SELECT count(distinct Salary) from WORKER b WHERE a.Salary <= b.Salary) order by a.Salary desc;

SELECT distinct Salary from WORKER a WHERE 3 >= (SELECT count(distinct Salary) from WORKER b WHERE a.Salary >= b.Salary) order by a.Salary desc;

SELECT distinct Salary from WORKER a WHERE 3 >= (SELECT count(distinct Salary) from WORKER b WHERE a.Salary <= b.Salary) order by a.Salary desc;
    
```

Results

SALARY
500000
300000
200000

3 rows returned in 0.00 seconds [Download](#)

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

47. Write An SQL Query To Fetch Three Min Salaries From Worker Table.

- SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary >= b.Salary) order by a.Salary desc;

```

SELECT DEPARTMENT, COUNT(WORKER_ID) as 'Number of Workers' FROM Worker GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;

SELECT FIRST_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY) from Worker);

SELECT DEPARTMENT, sum(Salary) from worker group by DEPARTMENT;

SELECT distinct Salary from worker a WHERE N >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;

SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary >= b.Salary) order by a.Salary desc;

```

Results

SALARY
90000
80000
75000

3 rows returned in 0.09 seconds Download

Application Express 4.0.2.00.09
Workspace: CSE User: CSE Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

48. Write An SQL Query To Fetch Nth Max Salaries From Worker Table.

- `SELECT DISTINCT SALARY FROM WORKER a WHERE 1>= (SELECT COUNT(DISTINCT SALARY) FROM WORKER b WHERE a.SALARY <= b.SALARY) ORDER BY a.SALARY DESC;`

```

SELECT SALARY FROM WORKER w1
WHERE 3-1 = (SELECT COUNT(DISTINCT SALARY) FROM WORKER w2
WHERE w2.SALARY>w1.SALARY);

SELECT * FROM WORKER INTERSECT SELECT * FROM WORKER_CLONE;

SELECT *
FROM WORKER
WHERE WORKER_ID <= (SELECT COUNT(WORKER_ID)/2 FROM WORKER);

SELECT DEPARTMENT, COUNT(WORKER_ID) FROM WORKER GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;

SELECT DEPARTMENT, COUNT(DEPARTMENT) AS 'NUMBER OF WORKERS' FROM WORKER GROUP BY DEPARTMENT;

SELECT DISTINCT SALARY FROM WORKER a WHERE 1>= (SELECT COUNT(DISTINCT SALARY) FROM WORKER b WHERE a.SALARY <= b.SALARY) ORDER BY a.SALARY DESC;

```

Results

SALARY
50000

1 rows returned in 0.04 seconds Download

Application Express 4.0.2.00.09
Workspace: CSE User: CSE Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

49. Write An SQL Query To Fetch Departments Along With The Total Salaries Paid For Each Of Them from Worker Table.

- SELECT DEPARTMENT, sum(Salary) from worker group by DEPARTMENT;

The screenshot shows the Oracle Application Express interface. In the SQL Commands tab, the following SQL code is entered:

```

(SELECT * FROM Worker)
INTERSECT
(SELECT * FROM WorkerClone);

SELECT DEPARTMENT, COUNT(WORKER_ID) as 'Number of Workers' FROM Worker GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;

SELECT FIRST_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY) from Worker);

SELECT DEPARTMENT, sum(Salary) from worker group by DEPARTMENT;

```

The Results section displays the output:

DEPARTMENT	SUM(SALARY)
HR	400000
Account	275000
Admin	1170000

3 rows returned in 0.00 seconds

50. Write An SQL Query To Fetch The Names Of Workers Who Earn The Highest Salary from Worker Table.

- SELECT FIRST_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY) from Worker);

The screenshot shows the Oracle Application Express interface. In the SQL Commands tab, the following SQL code is entered:

```

(SELECT * FROM Worker)
INTERSECT
(SELECT * FROM WorkerClone);

SELECT DEPARTMENT, COUNT(WORKER_ID) as 'Number of Workers' FROM Worker GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;

SELECT FIRST_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY) from Worker);

```

The Results section displays the output:

FIRST_NAME	SALARY
Vivek	500000
Amitabh	500000

2 rows returned in 0.03 seconds

ASSIGNMENT 7

25/11/2020 – 14/12/2020

1. Create the following tables with appropriate constraints using SQL command.

(a) Match

SLNO	Match_Id	Team1	Team2	Ground	Play_Date	Winner
------	----------	-------	-------	--------	-----------	--------

(b) Player

SLNO	Player_Id	Lname	Fname	Country	Yborn	Bplace	Ftest
------	-----------	-------	-------	---------	-------	--------	-------

(c) Bowling

SLNO	Match_Id	Player_Id	Novers	Maidens	Nrun_rev	Nwickets
------	----------	-----------	--------	---------	----------	----------

(d) Batting

SLNO	Match_Id	Player_Id	Nrun_sc
------	----------	-----------	---------

➤ create table Match_Cricket

(

SLNO number(3),

Match_id number(6) primary key,

Team1 varchar2(30),

Team2 varchar2(30),

Ground varchar2(30),

Play_date date,

Winner varchar2(30)

);

desc Match_Cricket;

The screenshot shows the Oracle Application Express 4.0.2 interface. In the top navigation bar, the path is Home > SQL Workshop > SQL Commands. The main area contains the following SQL code:

```
create table Match_Cricket
(
    SLNO number(3),
    Match_id number(6) primary key,
    Team1 varchar2(30),
    Team2 varchar2(30),
    Ground varchar2(30),
    Play_date date,
    Winner varchar2(30)
);

desc Match_Cricket;
create sequence SNO
    start with 1
    increment by 1
    nomaxvalue
    nocycle
```

Below the code, the 'Object Type' is shown as TABLE Object: MATCH_CRICKET. A detailed table structure is displayed:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MATCH_CRICKET	SLNO	NUMBER	-	3	0	-	✓	-	-
	MATCH_ID	NUMBER	-	6	0	1	-	-	-
	TEAM1	VARCHAR2	30	-	-	-	✓	-	-
	TEAM2	VARCHAR2	30	-	-	-	✓	-	-
	GROUND	VARCHAR2	30	-	-	-	✓	-	-
	PLAY_DATE	DATE	7	-	-	-	✓	-	-
	WINNER	VARCHAR2	30	-	-	-	✓	-	-

At the bottom right of the interface, it says 'Application Express 4.0.2 00.09'.

➤ create table Player_Cricket(

SLNO number(3),

Player_id number(10) primary key,

Lname varchar2(30),

Fname varchar2(30),

Country varchar2(30),

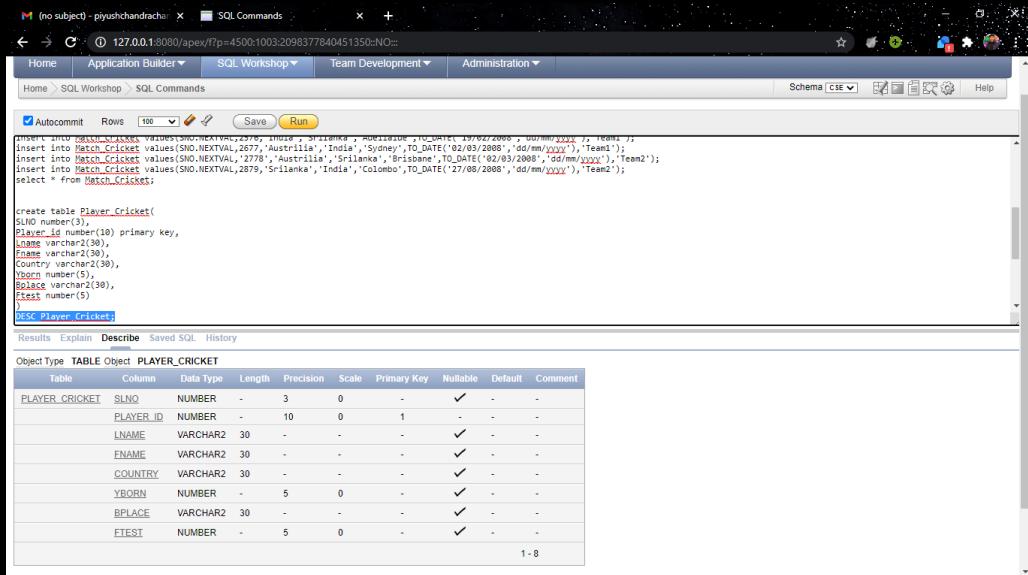
Yborn number(5),

Bplace varchar2(30),

Ftest number(5)

)

DESC Player_Cricket;



```

M (no subject) - piyushchandrabha... SQL Commands
127.0.0.1:8080/apex/r?=&f=45001003:2098377840451350:No...
Home Application Builder SQL Workshop Team Development Administration
Home SQL Workshop Schema CSE Help
Autocommit Rows 100 Save Run
insert into Match_Cricket values(2778, 'India', 'Mumbai', TO_DATE('28/07/2088', 'dd/mm/yyyy'), 'Team1');
insert into Match_Cricket values(2779, 'Australia', 'Sydney', TO_DATE('03/09/2088', 'dd/mm/yyyy'), 'Team1');
insert into Match_Cricket values(2780, 'Sri Lanka', 'Colombo', TO_DATE('02/05/2088', 'dd/mm/yyyy'), 'Team2');
insert into Match_Cricket values(2781, 'England', 'Edinburgh', TO_DATE('27/03/2088', 'dd/mm/yyyy'), 'Team1');
select * from Match_Cricket;

create table Player_Cricket(
SLNO number(3),
Player_id number(10) primary key,
Lname varchar2(30),
Fname varchar2(30),
Country varchar2(30),
Yborn number(5),
Bplace varchar2(30),
Ftest number(5)
);

```

Results Explain Describe Saved SQL History

Object Type	Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PLAYER_CRICKET	SLNO	NUMBER	-	3	0	-	✓	-	-	-
	PLAYER_ID	NUMBER	-	10	0	1	-	-	-	-
	LNAME	VARCHAR2	30	-	-	-	✓	-	-	-
	FNAME	VARCHAR2	30	-	-	-	✓	-	-	-
	COUNTRY	VARCHAR2	30	-	-	-	✓	-	-	-
	YBORN	NUMBER	-	5	0	-	✓	-	-	-
	BPLACE	VARCHAR2	30	-	-	-	✓	-	-	-
	FTEST	NUMBER	-	5	0	-	✓	-	-	-

➤ create table Bowling_Cricket(

SLNO Number(10),

Match_id number(10),

Player_id number(10),

Novers number(3),

Maidens number(3),

Nrun_rcv number(3),

Nwickets number(3),

constraint Fk_Match FOREIGN KEY(Match_id) references Match(Match_id),

constraint Fk_Player FOREIGN KEY(Player_id) references Player(Player_id)

)

desc Bowling_Cricket;

The screenshot shows the Oracle Application Express interface with the SQL Workshop tab selected. A SQL command window contains the following code:

```
SLNO Number(10),
Match_id number(10),
Player_id number(10),
Novers number(3),
Maidens number(3),
Nrun_rcv number(3),
Nwickets number(3),
constraint FK_Match FOREIGN KEY(Match_id) references Match(Match_id),
constraint FK_Player FOREIGN KEY(Player_id) references Player(Player_id)
);
desc Bowling_Cricket;
create sequence Ball
start with 1000
increment by 2
nomaxvalue
nocycle
cache 20;
```

Below the code, the 'Describe' tab is selected, showing the table structure:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BOWLING_CRICKET	SLNO	NUMBER	-	10	0	-	✓	-	-
	MATCH_ID	NUMBER	-	10	0	-	✓	-	-
	PLAYER_ID	NUMBER	-	10	0	-	✓	-	-
	NOVERS	NUMBER	-	3	0	-	✓	-	-
	MAIDENS	NUMBER	-	3	0	-	✓	-	-
	NRUN_RCV	NUMBER	-	3	0	-	✓	-	-
	NWICKETS	NUMBER	-	3	0	-	✓	-	-

At the bottom right of the interface, it says 'Application Express 4.0.2.0.09'.

➤ create table Batting_Cricket(

SLNO number(7),

Match_id number (10),

Player_id number(10),

Nrun_sc number(4),

constraint FK_player1 FOREIGN KEY(Player_id) references Player(Player_id),

constraint Fk_Match1 FOREIGN KEY(Match_id) references Match(Match_id)

)

desc Batting_Cricket;

The screenshot shows the Oracle Application Express interface with the SQL Workshop tab selected. A SQL command window contains the following code:

```
Insert into Bowling_Cricket values(Ball1,NEXTVAL,2376,490001,1,2,15,11);
Insert into Bowling_Cricket values(Ball1,NEXTVAL,2378,590001,7,1,20,5);
select * from Bowling_Cricket;

create table Batting_Cricket(
SLNO number(7),
Match_id number(10),
Player_id number(10),
constraint FK_player1 FOREIGN KEY(Player_id) references Player(Player_id),
constraint Fk_Match1 FOREIGN KEY(Match_id) references Match(Match_id)
);
desc Batting_Cricket;
create sequence Bat
start with 1000
increment by 2
nomaxvalue
nocycle
cache 20;
```

Below the code, the 'Describe' tab is selected, showing the table structure:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BATTING_CRICKET	SLNO	NUMBER	-	7	0	-	✓	-	-
	MATCH_ID	NUMBER	-	10	0	-	✓	-	-
	PLAYER_ID	NUMBER	-	10	0	-	✓	-	-
	NRUN_SC	NUMBER	-	4	0	-	✓	-	-

At the bottom right of the interface, it says 'Application Express 4.0.2.0.09'.

2. Add the following constraints

- BOWLING(Match_Id) references MATCH(Match_Id)
- BOWLING (Player_Id) references PLAYER(Player_Id)
- BATTING(Match_Id) references MATCH(Match_Id)
- BATTING (Player_Id) references PLAYER(Player_Id)
- Match (SLNO), Player (SLNO), Bowling (SLNO), Batting (SLNO): Auto Increment.

➤ **NOTE:-**

For Auto Increment you have to create sequence.|

➤ create sequence SNO

```
start with 1  
increment by 1  
nomaxvalue  
nocycle  
cache 20;
```

```
create sequence SP
```

```
start with 50  
increment by 10  
nomaxvalue  
nocycle  
cache 20;
```

```
create sequence Ball
```

```
start with 1000  
increment by 2  
nomaxvalue  
nocycle  
cache 20;
```

3. Insert the following data

(a) Data for Match table

SLNO	Match_Id	Team1	Team2	Ground	Play_Date	Winner
1	2475	Austrilia	India	Melbourn	10-FEB-08	Team2
2	2576	India	Srilanka	Adeliaide	19-FEB-08	Team1
3	2677	Austrilia	India	Sydney	02-MAR-08	Team1
4	2778	Austrilia	Srilanka	Brisbane	04-MAR-08	Team2
5	2879	Srilanka	India	Colombo	27-AUG-08	Team2

(a) Data for Bowling table

SLNO	Match_Id	Player_Id	Novers	Maidens	Nrun_rev	Nwickets
1000	2576	59001	8	3	58	1
1002	2677	69001	10	1	27	3
1004	2879	49002	7	0	44	0

1006	2576	49001	3	2	15	1
1008	2778	59001	7	1	20	5

(b) Data for Batting table

SLNO	Match_Id	Player_Id	Nrun_sc
501	2677	49001	60
502	2778	59002	71
503	2879	59001	60
504	2778	69002	17
505	2879	59002	45
506	2576	49001	36
507	2475	49002	72

(c) Data for Player table

SLNO	Player_Id	Lname	Fname	Country	Yborn	Bplace	Ftest
100	49001	Tendulkar	Sachin	India	1973	Mumbai	1986
90	49002	Dravid	Rahul	India	1973	Indore	1996
80	59001	Vass	Chaminda	Srilanka	1974	Mattumagal a	1994
70	59002	Jayasuriya	Sanath	Srilanka	1969	Matara	1991
60	69001	Lee	Brett	Austrilia	1976	Wollongong	1999
50	69002	Symonds	Andrew	Austrilia	1975	Birminghum	2004

- insert into Match_Cricket
values(SNO.NEXTVAL,2475,'Austrilia','India','Melbourn',TO_DATE('10/02/2008','dd/mm/yyyy'),'Team2');
- insert into Match_Cricket
values(SNO.NEXTVAL,2576,'India','Srilanka','Adeliaide',TO_DATE('19/02/2008','dd/mm/yyyy'),'Team1');
- insert into Match_Cricket
values(SNO.NEXTVAL,2677,'Austrilia','India','Sydney',TO_DATE('02/03/2008','dd/mm/yyyy'),'Team1');
- insert into Match_Cricket
values(SNO.NEXTVAL,2778,'Austrilia','Srilanka','Brisbane',TO_DATE('02/03/2008','dd/mm/yyyy'),'Team2');

```

insert into Match_Cricket
values(SNO.NEXTVAL,2879,'Srilanka','India','Colombo',TO_DATE('27/08/2008','dd/
mm/yyyy'),'Team2');

select * from Match_Cricket;

```

SQL Commands

```

desc Match_Cricket;
create sequence SNO
start with 1
increment by 1
nomaxvalue
cache 20;
Insert into Match_Cricket values(SNO.NEXTVAL,2475,'Australia','India','Melbourn',TO_DATE('18/02/2008','dd/mm/yyyy'),'Team2');
Insert into Match_Cricket values(SNO.NEXTVAL,2576,'India','Srilanka','Adelaide',TO_DATE('19/02/2008','dd/mm/yyyy'),'Team1');
Insert into Match_Cricket values(SNO.NEXTVAL,2677,'Australia','India','Sydney',TO_DATE('02/03/2008','dd/mm/yyyy'),'Team1');
Insert into Match_Cricket values(SNO.NEXTVAL,2778,'Australia','Srilanka','Brisbane',TO_DATE('02/03/2008','dd/mm/yyyy'),'Team2');
Insert into Match_Cricket values(SNO.NEXTVAL,2879,'Srilanka','India','Colombo',TO_DATE('27/08/2008','dd/mm/yyyy'),'Team2');
select * from Match_Cricket;

create table Player_Cricket;

```

Results

SLNO	MATCH_ID	TEAM1	TEAM2	GROUND	PLAY_DATE	WINNER
2	2475	Australia	India	Melbourn	02/10/2008	Team2
3	2576	India	Srilanka	Adelaide	02/19/2008	Team1
4	2677	Australia	India	Sydney	03/02/2008	Team1
5	2778	Australia	Srilanka	Brisbane	03/02/2008	Team2
6	2879	Srilanka	India	Colombo	08/27/2008	Team2

5 rows returned in 0.00 seconds

- insert into Player_Cricket
values(SP.NEXTVAL,69002,'Symonds','Andrew','Australiia',1975,'Birmingham','2004');
 - insert into Player_Cricket
values(SP.NEXTVAL,69001,'Lee','Brett','Australiia',1976,'Wollongong',1999);
 - insert into Player_Cricket
values(SP.NEXTVAL,59002,'Jayasuriya','Sanath','Srilanka',1969,'Matara',1991);
 - insert into Player_Cricket
values(SP.NEXTVAL,59001,'Vass','Chaminda','Srilanka',1974,'Mattumagala',1994);
 - insert into Player_Cricket
values(SP.NEXTVAL,49002,'Dravid','Rahul','India',1973,'Indore',1996);
 - insert into Player_Cricket
values(SP.NEXTVAL,49001,'Tendulkar','Sachin','India',1973,'Mumbai',1986);
- select * from Player_Cricket;

SQL Commands

```

desc varcarr(30);
desc number(5);
DESC Player_Cricket;
create sequence SP
start with 50
increment by 10
nomaxvalue
nocycle
nocache;
Insert into Player_Cricket values(SP.NEXTVAL,69002,'Symonds','Andrew','Australiia',1975,'Birmingham','2004');
Insert into Player_Cricket values(SP.NEXTVAL,69001,'Lee','Brett','Australiia',1976,'Wollongong',1999);
Insert into Player_Cricket values(SP.NEXTVAL,59002,'Jayasuriya','Sanath','Srilanka',1969,'Matara',1991);
Insert into Player_Cricket values(SP.NEXTVAL,59001,'Vass','Chaminda','Srilanka',1974,'Mattumagala',1994);
Insert into Player_Cricket values(SP.NEXTVAL,49002,'Dravid','Rahul','India',1973,'Indore',1996);
Insert into Player_Cricket values(SP.NEXTVAL,49001,'Tendulkar','Sachin','India',1973,'Mumbai',1986);
select * from Player_Cricket;

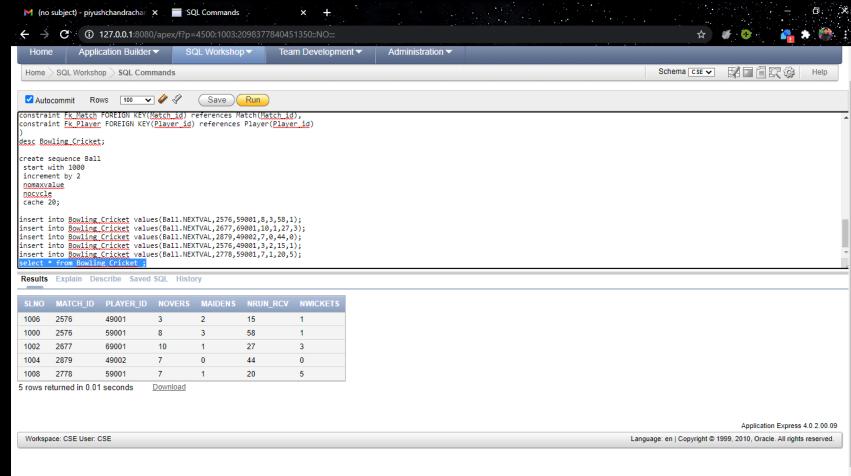
```

Results

SLNO	PLAYER_ID	LNAME	FNAME	COUNTRY	YBORN	BIPLACE	FTTEST
90	59001	Vass	Chaminda	Srilanka	1974	Mattumagala	1994
60	69002	Symonds	Andrew	Australiia	1975	Birmingham	2004
70	69001	Lee	Brett	Australiia	1976	Wollongong	1999
80	59002	Jayasuriya	Sanath	Srilanka	1969	Matara	1991
100	49002	Dravid	Rahul	India	1973	Indore	1996
110	49001	Tendulkar	Sachin	India	1973	Mumbai	1986

6 rows returned in 0.05 seconds

- insert into Bowling_Cricket values(Ball.NEXTVAL,2576,59001,8,3,58,1);
 insert into Bowling_Cricket values(Ball.NEXTVAL,2677,69001,10,1,27,3);
 insert into Bowling_Cricket values(Ball.NEXTVAL,2879,49002,7,0,44,0);
 insert into Bowling_Cricket values(Ball.NEXTVAL,2576,49001,3,2,15,1);
 insert into Bowling_Cricket values(Ball.NEXTVAL,2778,59001,7,1,20,5);
 select * from Bowling_Cricket;



```

SQL Commands
Home Application Builder SQL Workshop Team Development Administration
Autocommit Rows 100 Save Run
constraint FK_Match FOREIGN KEY(Match_Id) REFERENCES Match(Match_Id);
constraint FK_Player FOREIGN KEY(Player_Id) REFERENCES Player(Player_Id);

desc Bowling_Cricket;
create sequence Ball
start with 1000
increment by 2
nomaxvalue
nocycle;
cache 20;
Insert into Bowling_Cricket values(Ball.NEXTVAL,2576,59001,8,3,58,1);
Insert into Bowling_Cricket values(Ball.NEXTVAL,2677,69001,10,1,27,3);
Insert into Bowling_Cricket values(Ball.NEXTVAL,2879,49002,7,0,44,0);
Insert into Bowling_Cricket values(Ball.NEXTVAL,2576,49001,3,2,15,1);
Insert into Bowling_Cricket values(Ball.NEXTVAL,2778,59001,7,1,20,5);
select * from Bowling_Cricket;

```

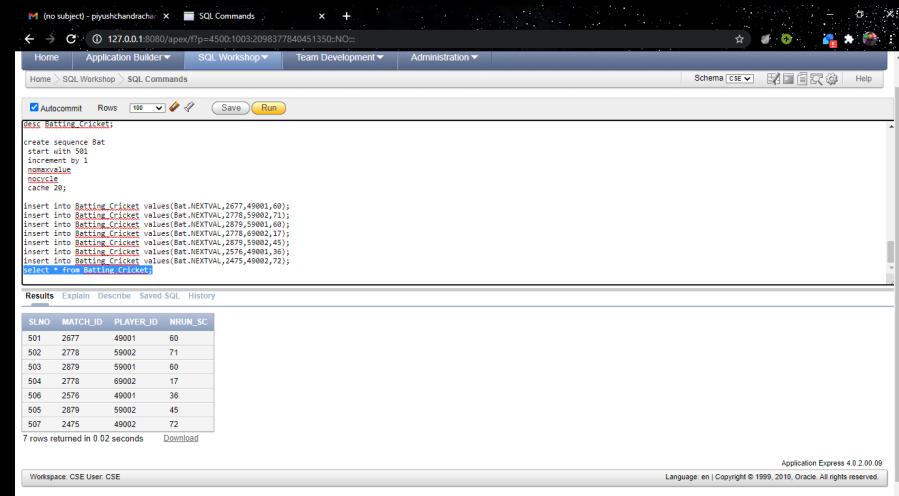
Results

S.L.NO	MATCH_ID	PLAYER_ID	NOVERS	MAIDENS	NRUN_NCV	NWICKETS
1006	2576	49001	3	2	15	1
1000	2576	59001	8	3	58	1
1002	2677	69001	10	1	27	3
1004	2879	49002	7	0	44	0
1008	2778	59001	7	1	20	5

5 rows returned in 0.01 seconds Download

Application Express 4.0.2.00.09
Workspace: CSE User: CSE Language: en Copyright © 1999, 2010, Oracle. All rights reserved.

- insert into Batting_Cricket values(Bat.NEXTVAL,2677,49001,60);
 insert into Batting_Cricket values(Bat.NEXTVAL,2778,59002,71);
 insert into Batting_Cricket values(Bat.NEXTVAL,2879,59001,60);
 insert into Batting_Cricket values(Bat.NEXTVAL,2778,69002,17);
 insert into Batting_Cricket values(Bat.NEXTVAL,2879,59002,45);
 insert into Batting_Cricket values(Bat.NEXTVAL,2576,49001,36);
 insert into Batting_Cricket values(Bat.NEXTVAL,2475,49002,72);
 select * from Batting_Cricket;



```

SQL Commands
Home Application Builder SQL Workshop Team Development Administration
Autocommit Rows 100 Save Run
desc Batting_Cricket;
create sequence Bat
start with 501
increment by 1
nomaxvalue
nocycle;
cache 20;
Insert into Batting_Cricket values(Bat.NEXTVAL,2677,49001,60);
Insert into Batting_Cricket values(Bat.NEXTVAL,2778,59002,71);
Insert into Batting_Cricket values(Bat.NEXTVAL,2879,59001,60);
Insert into Batting_Cricket values(Bat.NEXTVAL,2778,69002,17);
Insert into Batting_Cricket values(Bat.NEXTVAL,2879,59002,45);
Insert into Batting_Cricket values(Bat.NEXTVAL,2576,49001,36);
Insert into Batting_Cricket values(Bat.NEXTVAL,2475,49002,72);
select * from Batting_Cricket;

```

Results

S.L.NO	MATCH_ID	PLAYER_ID	NRUN_SC
501	2677	49001	60
502	2778	59002	71
503	2879	59001	60
504	2778	69002	17
506	2576	49001	36
505	2879	59002	45
507	2475	49002	72

7 rows returned in 0.02 seconds Download

Application Express 4.0.2.00.09
Workspace: CSE User: CSE Language: en Copyright © 1999, 2010, Oracle. All rights reserved.

- 4.
- Find the ground of the matches batted by player whose Fname is starting from 'S'.
 - Find Id's of player that have bowled in the match 2576 but not have batted.
 - Find the batting average of each Indian player along with the Player_Id
 - Find the name of that player who has bowled highest number of overs and also find the ground where he has bowled.

a) select Match_Cricket.Ground,Player_Cricket.Fname from Match_Cricket,Player_Cricket,Batting_Cricket

where Match_Cricket.Match_id=Batting_Cricket.Match_id and Batting_Cricket.Player_id=Player_Cricket.Player_id

and Player_Cricket.Fname like 'S%';

```

RECYCLE
cache 20;
Insert into Batting_Cricket values(Bat.NEXTVAL,2677,49001,60);
Insert into Batting_Cricket values(Bat.NEXTVAL,2778,59002,71);
Insert into Batting_Cricket values(Bat.NEXTVAL,2879,59001,60);
Insert into Batting_Cricket values(Bat.NEXTVAL,2979,59002,45);
Insert into Batting_Cricket values(Bat.NEXTVAL,2879,59002,45);
Insert into Batting_Cricket values(Bat.NEXTVAL,2576,49001,36);
Insert into Batting_Cricket values(Bat.NEXTVAL,2475,49002,72);
Select * From Batting_Cricket;
select Match_Cricket.Ground,Player_Cricket.Fname from Match_Cricket,Player_Cricket,Batting_Cricket
where Match_Cricket.Match_id=Batting_Cricket.Match_id and Batting_Cricket.Player_id=Player_Cricket.Player_id
and Player_Cricket.Fname like 'S%';

```

Results

GROUND	NAME
Sydney	Sachin
Brisbane	Sanath
Adelaide	Sachin
Colombo	Sanath

4 rows returned in 0.08 seconds

b) SELECT PLAYER_ID
 FROM Bowling_Cricket
 WHERE MATCH_ID = 2576
 AND PLAYER_ID NOT IN(SELECT PLAYER_ID
 FROM Batting_Cricket
 WHERE MATCH_ID = 2576);

```

Insert into Batting_Cricket values(Bat.NEXTVAL,2677,49001,60);
Insert into Batting_Cricket values(Bat.NEXTVAL,2778,59002,71);
Insert into Batting_Cricket values(Bat.NEXTVAL,2879,59001,60);
Insert into Batting_Cricket values(Bat.NEXTVAL,2979,59002,45);
Insert into Batting_Cricket values(Bat.NEXTVAL,2879,59002,45);
Insert into Batting_Cricket values(Bat.NEXTVAL,2576,49001,36);
Insert into Batting_Cricket values(Bat.NEXTVAL,2475,49002,72);
Select * From Batting_Cricket;
select Match_Cricket.Ground,Player_Cricket.Fname from Match_Cricket,Player_Cricket,Batting_Cricket
where Match_Cricket.Match_id=Batting_Cricket.Match_id and Batting_Cricket.Player_id=Player_Cricket.Player_id
and Player_Cricket.Fname like 'S%';
SELECT PLAYER_ID
FROM Bowling_Cricket
WHERE MATCH_ID = 2576
AND PLAYER_ID NOT IN(SELECT PLAYER_ID
FROM Batting_Cricket
WHERE MATCH_ID = 2576);

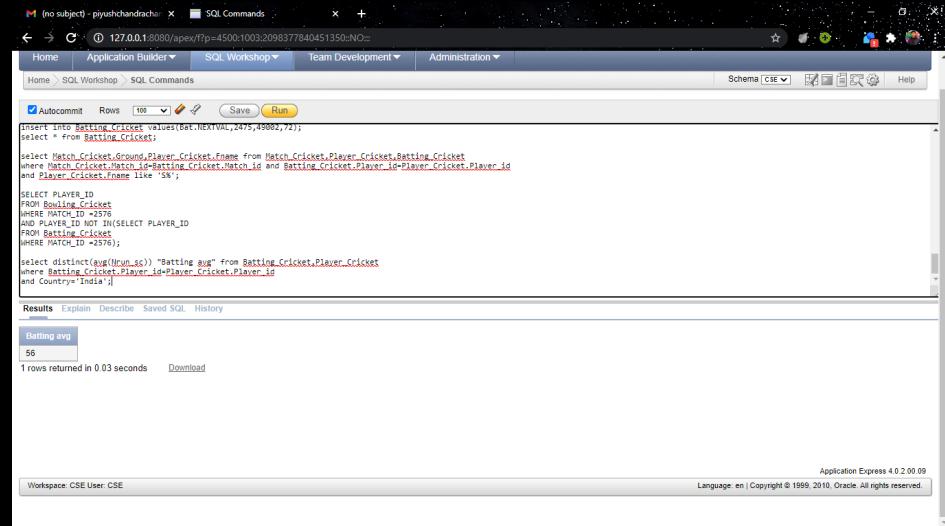
```

Results

PLAYER_ID
59001

1 rows returned in 0.09 seconds

- c) select distinct(avg(Nrun_sc)) "Batting avg" from Batting_Cricket,Player_Cricket
 where Batting_Cricket.Player_id=Player_Cricket.Player_id
 and Country='India';



```

M (no subject) - piyushchandrar  SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:2098377840451350:N
Home Application Builder SQL Workshop Team Development Administration
Home SQL Workshop SQL Commands Schema CSE Help
Autocommit Rows 100 Save Run
insert into Batting_Cricket values(BAT.RECTVAL,245,49002,72);
select * from Batting_Cricket;
select Match_Cricket.Ground,Player_Cricket.Fname from Match_Cricket,Player_Cricket,Batting_Cricket
where Match_Cricket.Match_id=Batting_Cricket.Match_id and Batting_Cricket.Player_id=Player_Cricket.Player_id
and Player_Cricket.Fname like 'SN';
SELECT PLAYER_ID
FROM Bowling_Cricket
WHERE MATCH_ID < 2576
AND PLAYER_ID NOT IN(SELECT PLAYER_ID
FROM Batting_Cricket
WHERE MATCH_ID > 2576);
select distinct(avg(Nrun_sc)) "Batting avg" from Batting_Cricket,Player_Cricket
where Batting_Cricket.Player_id=Player_Cricket.Player_id
and Country='India';
  
```

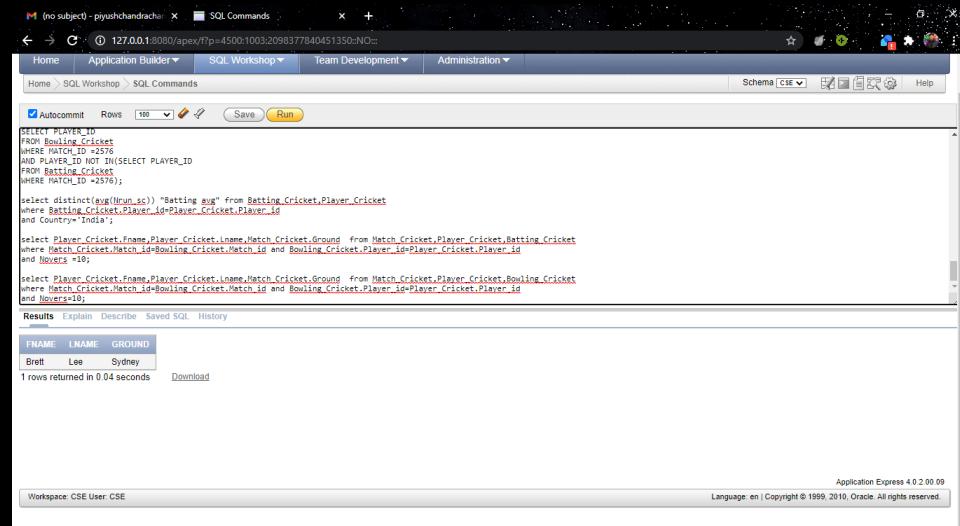
Results Explain Describe Saved SQL History

Batting avg
56

1 rows returned in 0.03 seconds Download

Application Express 4.0.2.00.09
Language: en | Copyright © 1999-2010, Oracle. All rights reserved.

- d) select Player_Cricket.Fname,Player_Cricket.Lname,Match_Cricket.Ground from Match_Cricket,Player_Cricket,Bowling_Cricket
 where Match_Cricket.Match_id=Bowling_Cricket.Match_id and
 Bowling_Cricket.Player_id=Player_Cricket.Player_id
 and Novers=10;



```

M (no subject) - piyushchandrar  SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:2098377840451350:N
Home Application Builder SQL Workshop Team Development Administration
Home SQL Workshop SQL Commands Schema CSE Help
Autocommit Rows 100 Save Run
SELECT PLAYER_ID
FROM Bowling_Cricket
WHERE MATCH_ID < 10
AND PLAYER_ID NOT IN(SELECT PLAYER_ID
FROM Batting_Cricket
WHERE MATCH_ID > 2576);
select distinct(avg(Nrun_sc)) "Batting avg" from Batting_Cricket,Player_Cricket
where Batting_Cricket.Player_id=Player_Cricket.Player_id
and Country='India';
select Player_Cricket.Fname,Player_Cricket.Lname,Match_Cricket.Ground from Match_Cricket,Player_Cricket,Batting_Cricket
where Match_Cricket.Match_id=Bowling_Cricket.Match_id and Bowling_Cricket.Player_id=Player_Cricket.Player_id
and Novers=10;
select Player_Cricket.Fname,Player_Cricket.Lname,Match_Cricket.Ground from Match_Cricket,Player_Cricket,Bowling_Cricket
where Match_Cricket.Match_id=Bowling_Cricket.Match_id and Bowling_Cricket.Player_id=Player_Cricket.Player_id
and Novers=10;
  
```

Results Explain Describe Saved SQL History

FNAME	LNAME	GROUND
Brett	Lee	Sydney

1 rows returned in 0.04 seconds Download

Application Express 4.0.2.00.09
Language: en | Copyright © 1999-2010, Oracle. All rights reserved.

5.

- a) Find the total run scored by a player who played First Test in 1991.
- b) Find the name and the number of wickets taken by the youngest player in the data base.
- c) Find the names of the players who batted in only one match.
- d) Find the name of the player who has taken the highest wickets in a particular match and also find the ground where the player has taken the wickets.

a) select sum(nrun_sc) as Total_runs from Batting_Cricket

where player_id in (select player_id from Player_Cricket where ftest=1991);

The screenshot shows the Oracle Application Express interface with the SQL Workshop tab selected. The code area contains the following SQL query:

```
select player_id, avg(nrun_sc) as average_batting from batting
where player_id in (select player_id from player where country like 'India')
group by player_id

/So/
select sum(nrun_sc) as total_runs from Batting_Cricket
where player_id in (select player_id from Player_Cricket where ftest=1991);
```

The results section shows a single row with the value 116 under the column TOTAL_RUNS. The status bar at the bottom right indicates "Application Express 4.0.2.00.09".

b) select Player_Cricket.fname,Player_Cricket.lname,Bowling_Cricket.nwickets
from Player_Cricket,Bowling_Cricket

where Player_Cricket.player_id = Bowling_Cricket.player_id

and Player_Cricket.yborn = (select max(yborn) from Player_Cricket);

The screenshot shows the Oracle Application Express interface with the SQL Workshop tab selected. The code area contains the following SQL query:

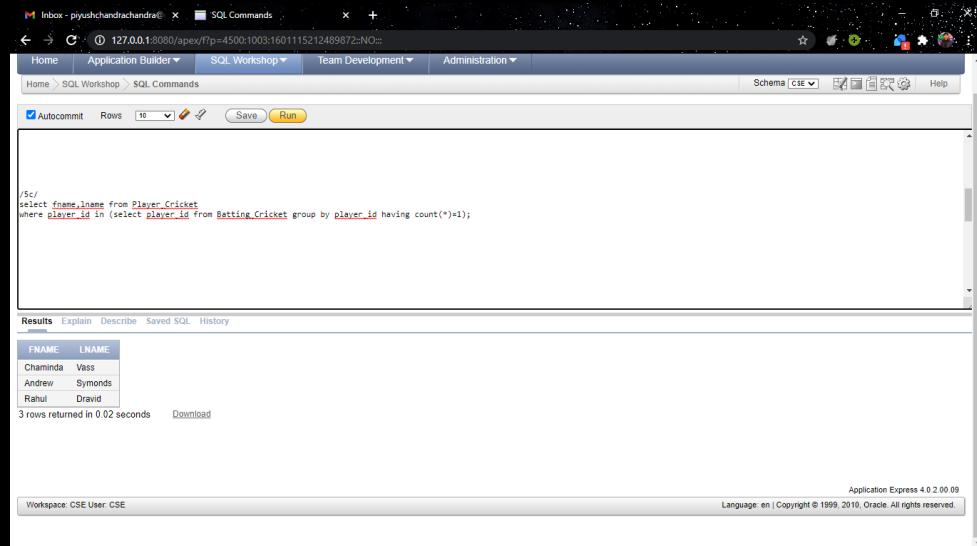
```
/So/
select Player_Cricket.fname,Player_Cricket.lname,Bowling_Cricket.nwickets from Player_Cricket,Bowling_Cricket
where Player_Cricket.player_id = Bowling_Cricket.player_id
and Player_Cricket.yborn = (select max(yborn) from Player_Cricket);


```

The results section shows a single row with the values Brett, Lee, and 3 under the columns FNAME, LNAME, and NWICKETS respectively. The status bar at the bottom right indicates "Application Express 4.0.2.00.09".

c) select fname,lname from Player_Cricket

where player_id in (select player_id from Batting_Cricket group by player_id having count(*)=1);



The screenshot shows the Oracle Application Express SQL Workshop interface. The query entered is:

```
/S/
select fname,lname from Player_Cricket
where player_id in (select player_id from Batting_Cricket group by player_id having count(*)=1);
```

The results table shows three rows:

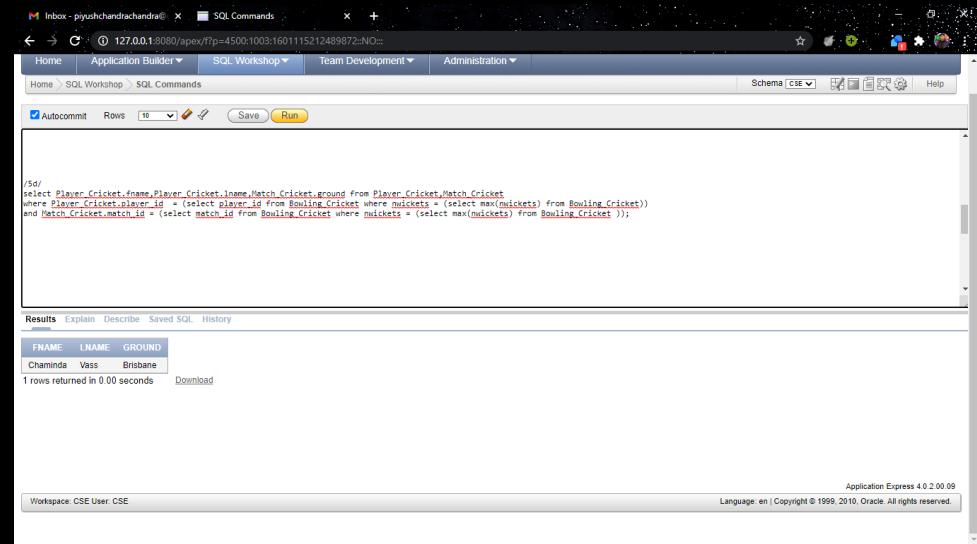
FNAME	LNAME
Chaminda	Vass
Andrew	Symonds
Rahul	Dravid

3 rows returned in 0.02 seconds

d) select Player_Cricket.fname,Player_Cricket.lname,Match_Cricket.ground from Player_Cricket,Match_Cricket

where Player_Cricket.player_id = (select player_id from Bowling_Cricket where nwickets = (select max(nwickets) from Bowling_Cricket))

and Match_Cricket.match_id = (select match_id from Bowling_Cricket where nwickets = (select max(nwickets) from Bowling_Cricket));



The screenshot shows the Oracle Application Express SQL Workshop interface. The query entered is:

```
/Sd/
select Player_Cricket.fname,Player_Cricket.lname,Match_Cricket.ground from Player_Cricket,Match_Cricket
where Player_Cricket.player_id = (select player_id from Bowling_Cricket where nwickets = (select max(nwickets) from Bowling_Cricket))
and Match_Cricket.match_id = (select match_id from Bowling_Cricket where nwickets = (select max(nwickets) from Bowling_Cricket));
```

The results table shows one row:

FNAME	LNAME	GROUND
Chaminda	Vass	Brisbane

1 rows returned in 0.00 seconds

6.

- a) Find the ground where Sachin Tendulkar has scored his highest run.
- b) Find out the name of a Srilankan bowler who has delivered at least 2 maiden overs.
- c) Find the Number of wickets of that player whose Birth place in Mattumagala.
- d) Find the names of the players who played in more than one matches.

a) select Match_Cricket.ground from Match_Cricket

inner join Batting_Cricket

on Match_Cricket.match_id = Batting_Cricket.match_id

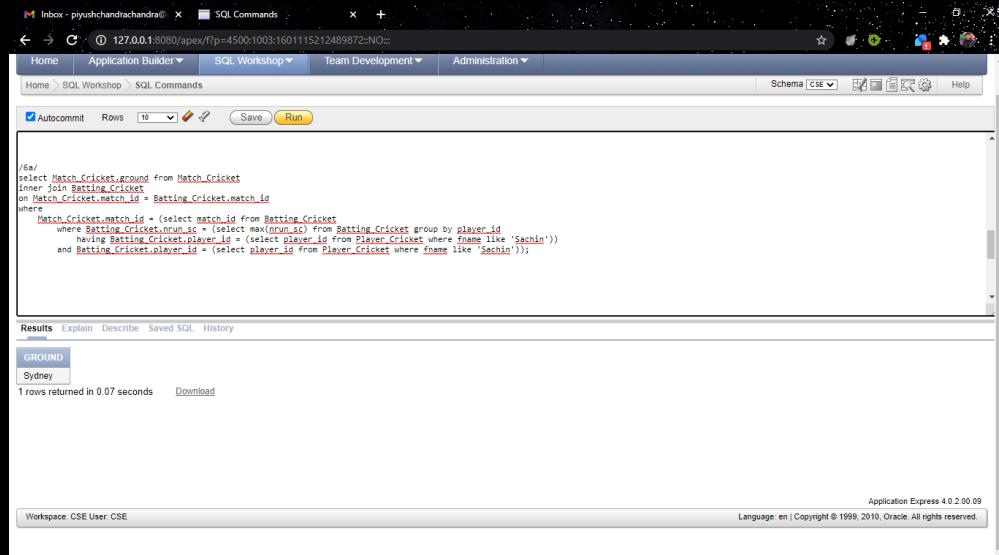
where

Match_Cricket.match_id = (select match_id from Batting_Cricket

where Batting_Cricket.nrun_sc = (select max(nrun_sc) from
Batting_Cricket group by player_id

having Batting_Cricket.player_id = (select player_id from Player_Cricket
where fname like 'Sachin'))

and Batting_Cricket.player_id = (select player_id from Player_Cricket
where fname like 'Sachin'));



The screenshot shows the Oracle Application Express SQL Workshop interface. The query window contains the following SQL code:

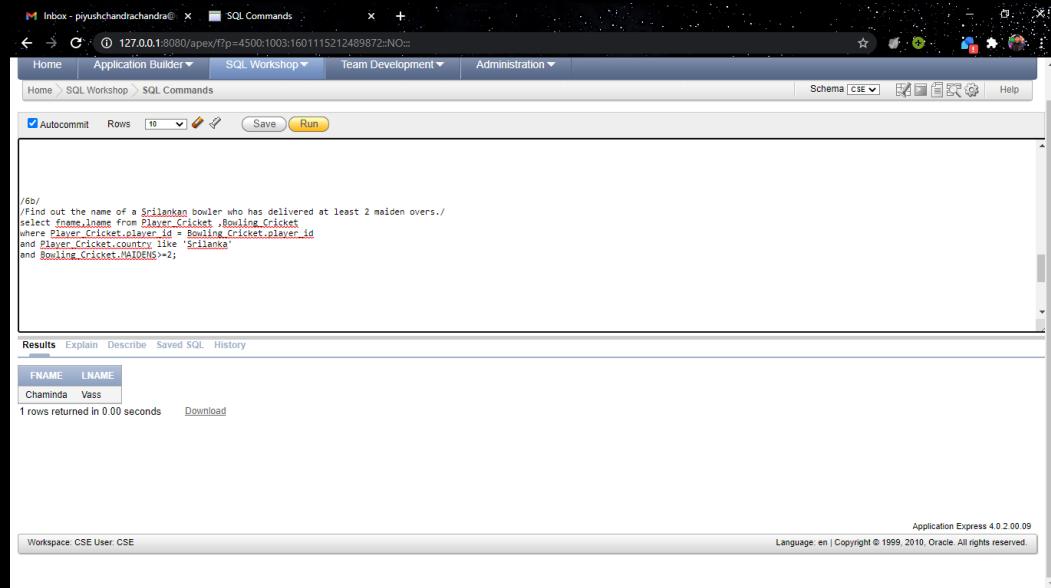
```
/6a/
select Match_Cricket.ground from Match_Cricket
inner join Batting_Cricket
on Match_Cricket.match_id = Batting_Cricket.match_id
where
    Match_Cricket.match_id = (select match_id from Batting_Cricket
        where Batting_Cricket.nrun_sc = (select max(nrun_sc) from Batting_Cricket group by player_id
            having Batting_Cricket.player_id = (select player_id from Player_Cricket where fname like 'Sachin'))
        and Batting_Cricket.player_id = (select player_id from Player_Cricket where fname like 'Sachin'));
```

The results section shows a single row:

GROUND
Sydney

1 rows returned in 0.07 seconds

b) select fname,lname from Player_Cricket ,Bowling_Cricket
 where Player_Cricket.player_id = Bowling_Cricket.player_id
 and Player_Cricket.country like 'Srilanka'
 and Bowling_Cricket.MAIDENS>=2;



```

SQL
--Find out the name of a Srilankan bowler who has delivered at least 2 maiden overs.
select fname,lname from Player_Cricket,Bowling_Cricket
where Player_Cricket.player_id = Bowling_Cricket.player_id
and Player_Cricket.country like 'Srilanka'
and Bowling_Cricket.MAIDENS>=2;
  
```

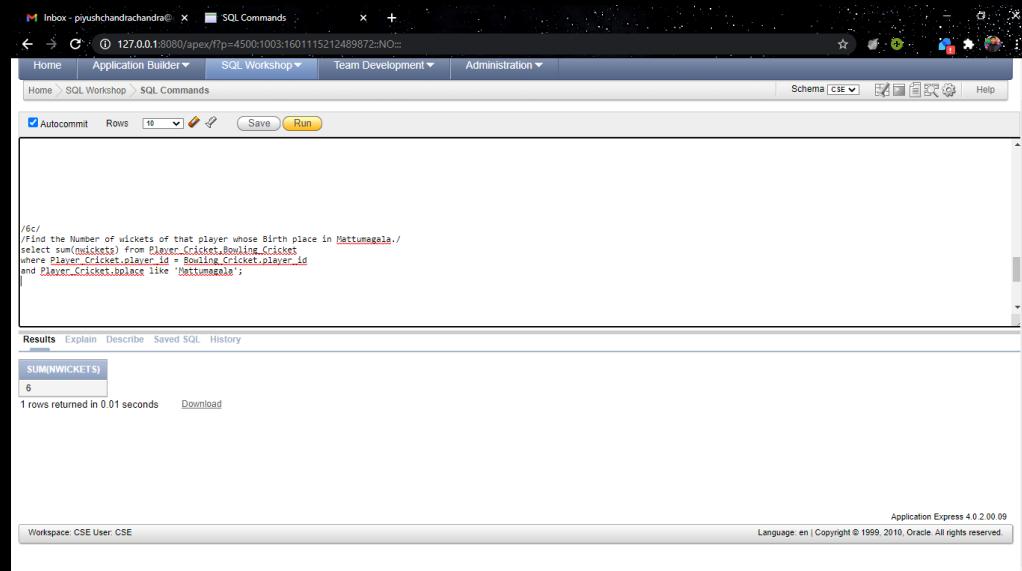
Results

FNAME	LNAME
Chaminda	Vass

1 rows returned in 0.00 seconds Download

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

c) select sum(nwickets) from Player_Cricket,Bowling_Cricket
 where Player_Cricket.player_id = Bowling_Cricket.player_id
 and Player_Cricket.bplace like 'Mattumagala';



```

SQL
--Find the Number of wickets of that player whose Birth place in Mattumagala.
select sum(nwickets) from Player_Cricket,Bowling_Cricket
where Player_Cricket.player_id = Bowling_Cricket.player_id
and Player_Cricket.bplace like 'Mattumagala';
  
```

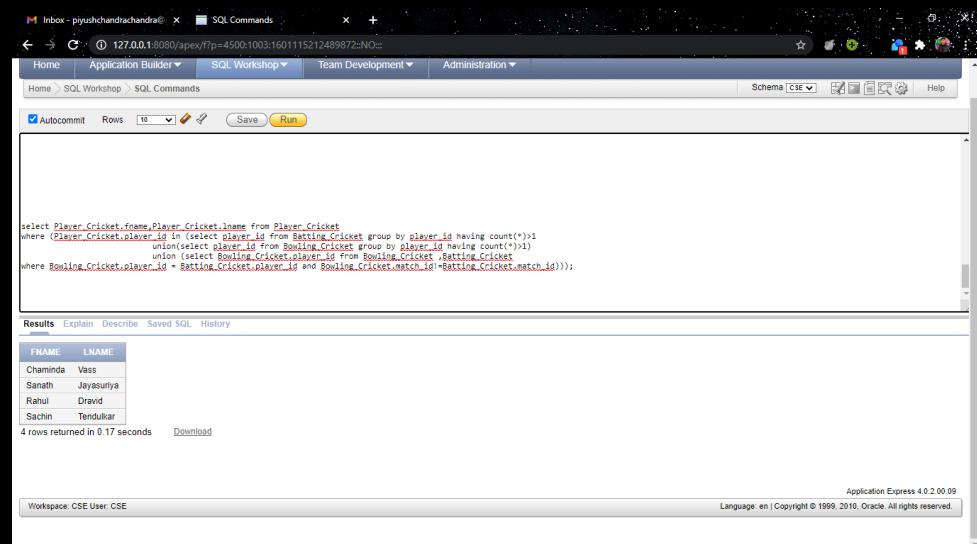
Results

SUM(NWICKETS)
6

1 rows returned in 0.01 seconds Download

Application Express 4.0.2.00.09
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

d) select Player_Cricket.fname,Player_Cricket.lname from Player_Cricket
 where (Player_Cricket.player_id in (select player_id from Batting_Cricket group by player_id having count(*)>1
 union(select player_id from Bowling_Cricket group by player_id having count(*)>1)
 union (select Bowling_Cricket.player_id from Bowling_Cricket ,Batting_Cricket
 where Bowling_Cricket.player_id = Batting_Cricket.player_id and Bowling_Cricket.match_id!=Batting_Cricket.match_id));



```

select Player_Cricket.fname,Player_Cricket.lname from Player_Cricket
where (Player_Cricket.player_id in (select player_id from Batting_Cricket group by player_id having count(*)>1
union(select player_id from Bowling_Cricket group by player_id having count(*)>1)
union (select Bowling_Cricket.player_id from Bowling_Cricket ,Batting_Cricket
where Bowling_Cricket.player_id = Batting_Cricket.player_id and Bowling_Cricket.match_id!=Batting_Cricket.match_id)));
  
```

FNAME	LNAME
Chaminda	Vass
Sanath	Jayasuriya
Rahul	Dravid
Sachin	Tendulkar

4 rows returned In 0.17 seconds [Download](#)

THANK YOU

~~~~~ X ~~~~~