SISTER NIVEDITA UNIVERSITY

DATABASE MANAGEMENT SYSTEM

SUBMITTED BY: PIYUSH CHANDRA CHANDRA

DEPT- BTECH (CSE), ROLL- 1027

SUBMITTED TO: SWARUP KUMAR GHOSH &

DEBANJAN DAS

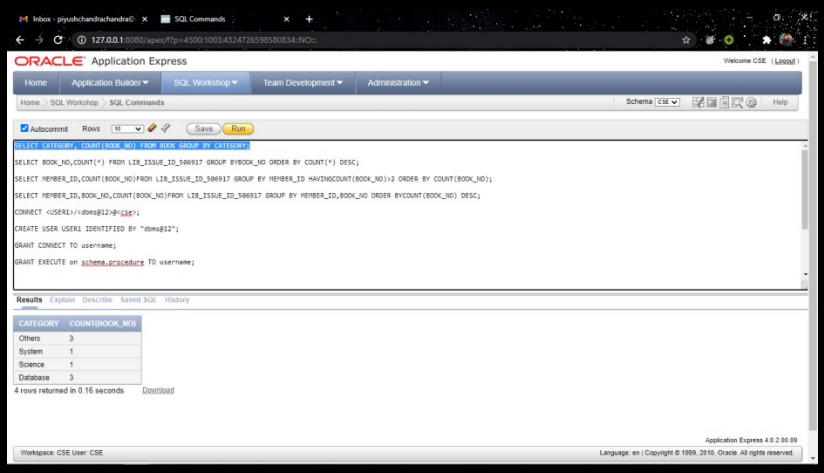
INDEX

ASSIGNMENT 5

10/11/2020 – 17/11/2020

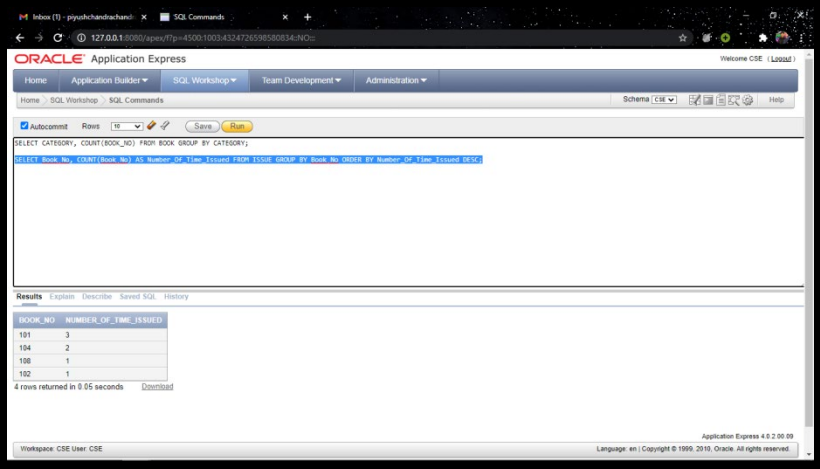
1. List the various CATEGORIES and COUNT OF BOOKS in each category.

- SELECT CATEGORY, COUNT(BOOK_NO) FROM BOOK GROUP BY CATEGORY;



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



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

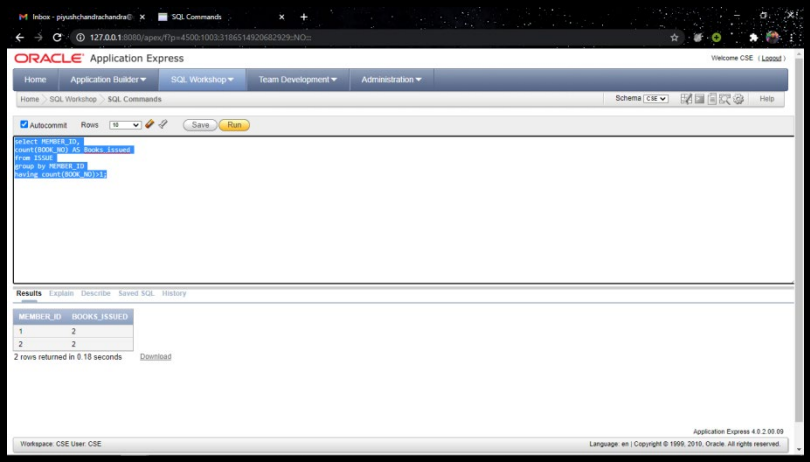
- select MEMBER_ID,

  count(BOOK_NO) AS Books_issued

  from ISSUE

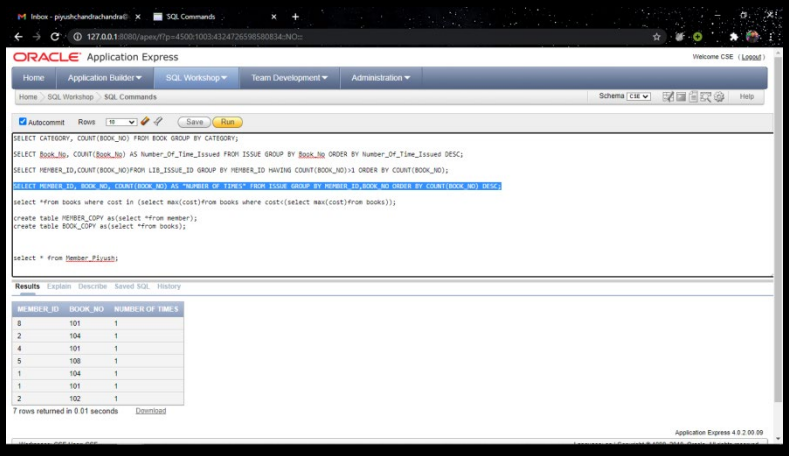
  group by MEMBER_ID

  having count(BOOK_NO)>1;



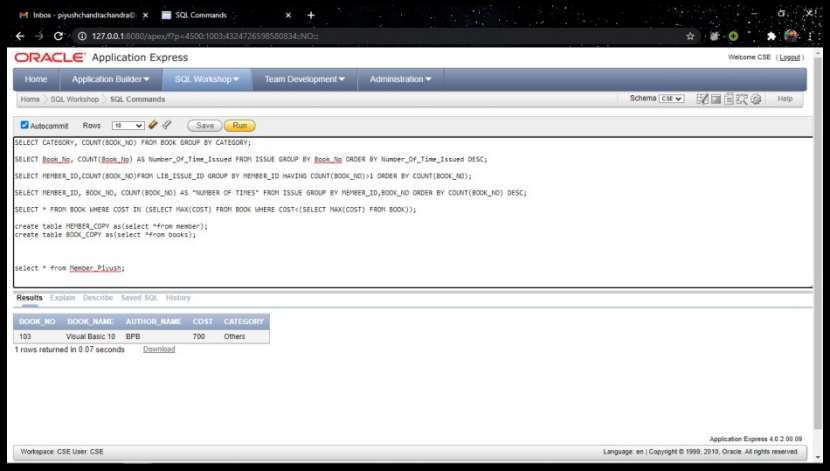
4. Display the MEMBER ID, BOOK NO and NO OF TIMES the same book is issued by the MEMBER in the descending order of count.

- 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;



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));



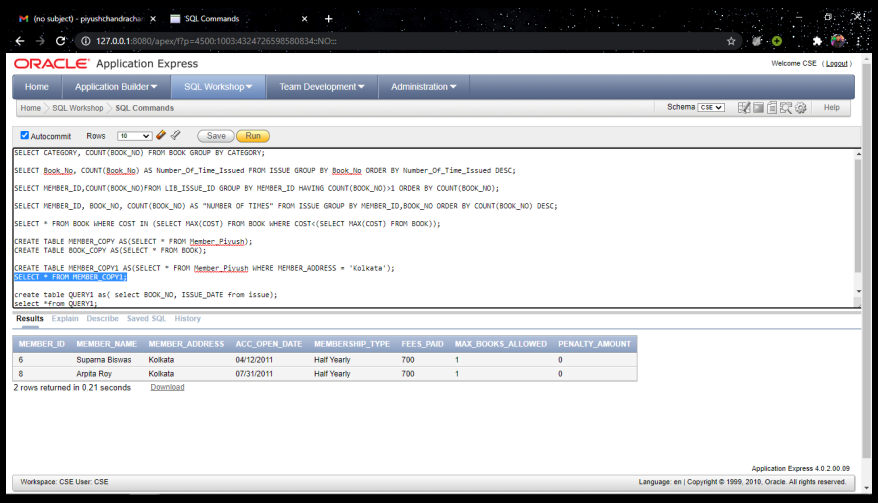
6.Copy the MEMBER, BOOKS and ISSUE tables and MEMBER_COPY, BOOKS_COPY, ISSUE_COPY respectively.

- 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');

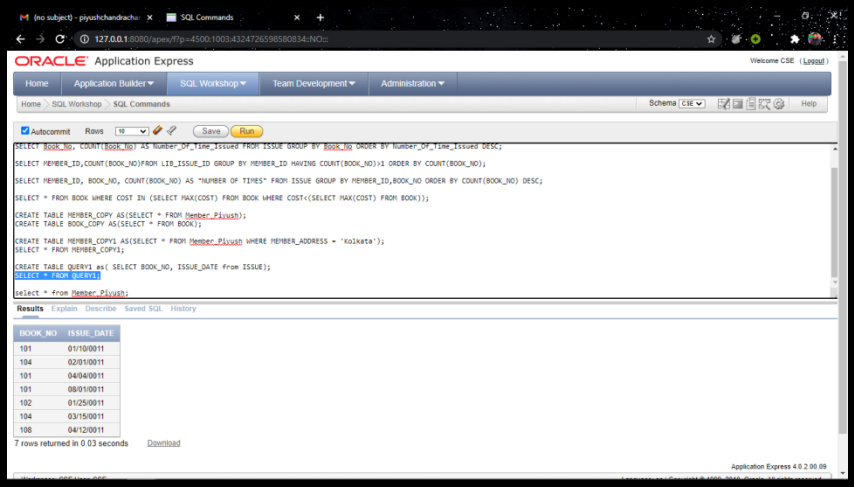
  SELECT * FROM MEMBER_COPY1;



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.

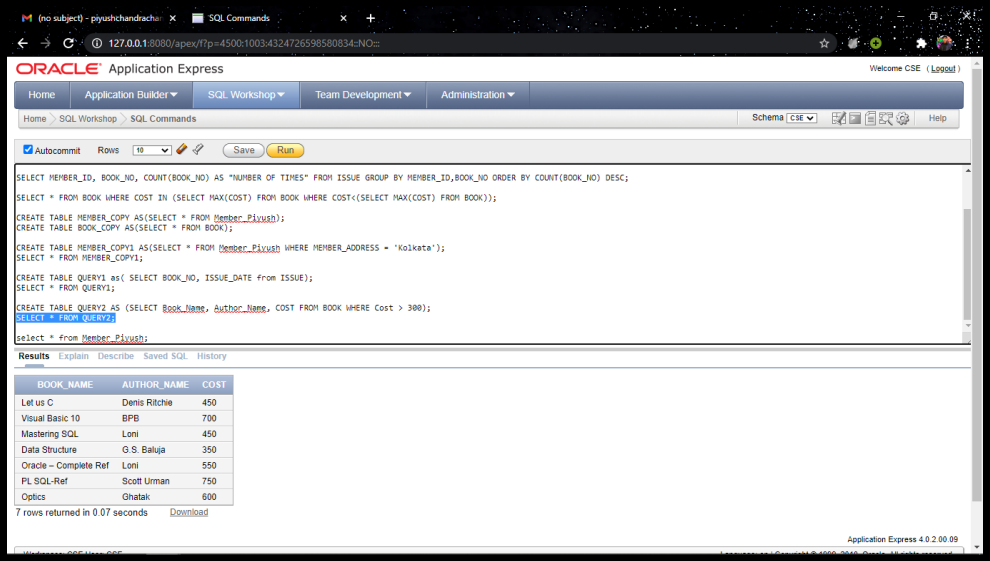
➢ CREATE TABLE QUERY1 as(SELECT BOOK_NO, ISSUE_DATE from ISSUE);

SELECT * FROM QUERY1;



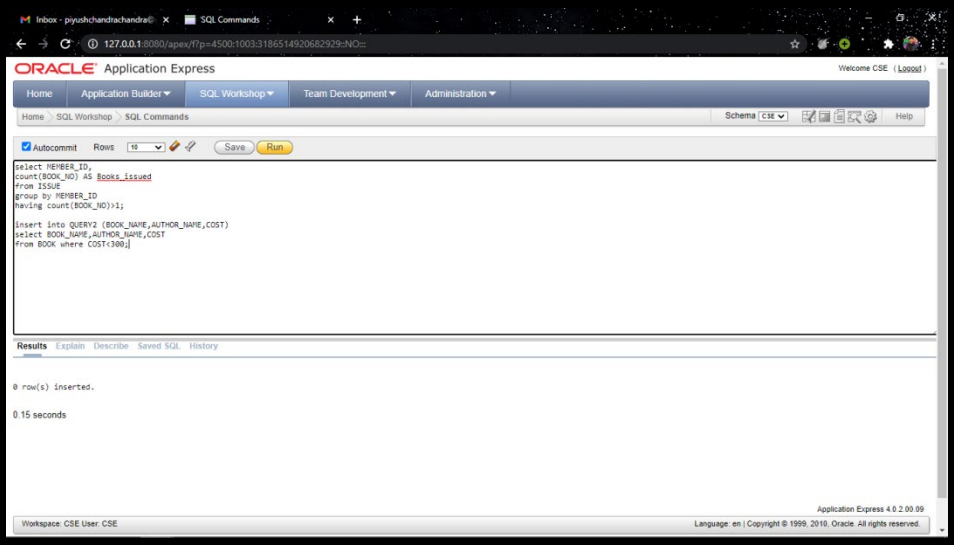
8. Create and store BOOK_NAME, AUTHOR_NAME and COST to QUERY2 table where cost is greater than 300.

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



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;



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;

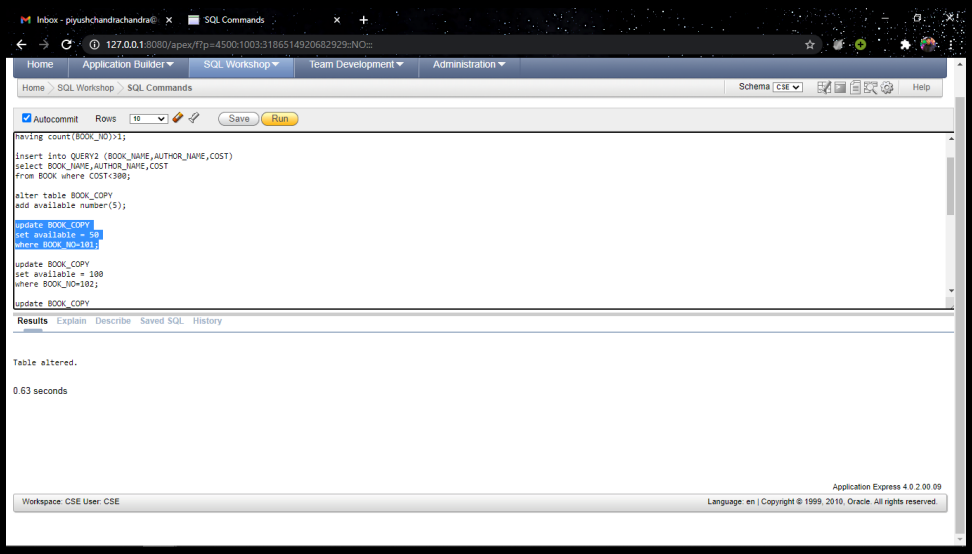
  update BOOK_COPY

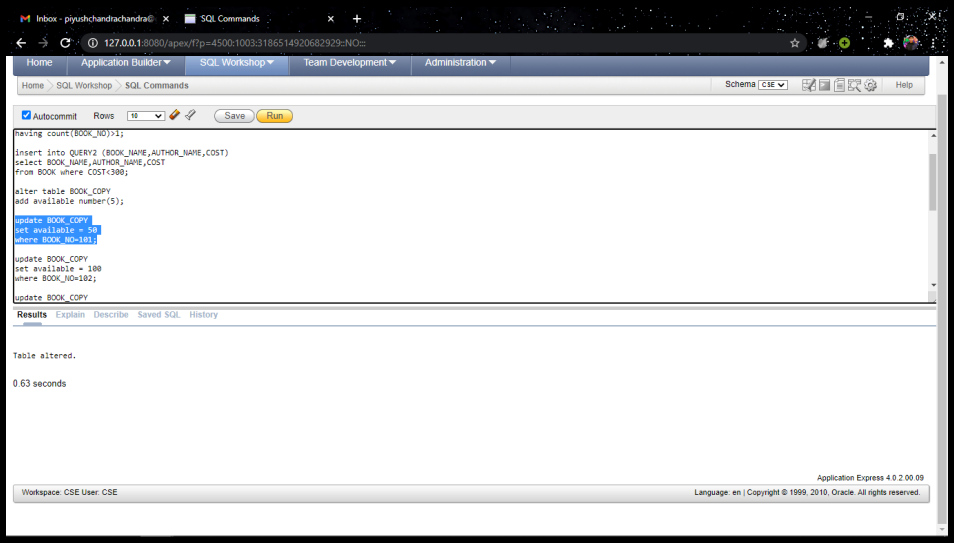
  set available = 100

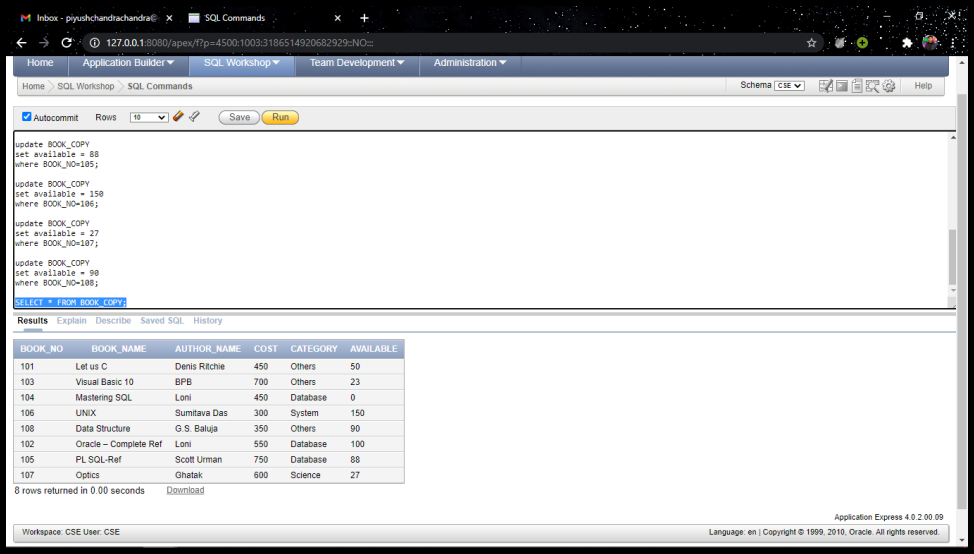
  where BOOK_NO=102;

  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;
```

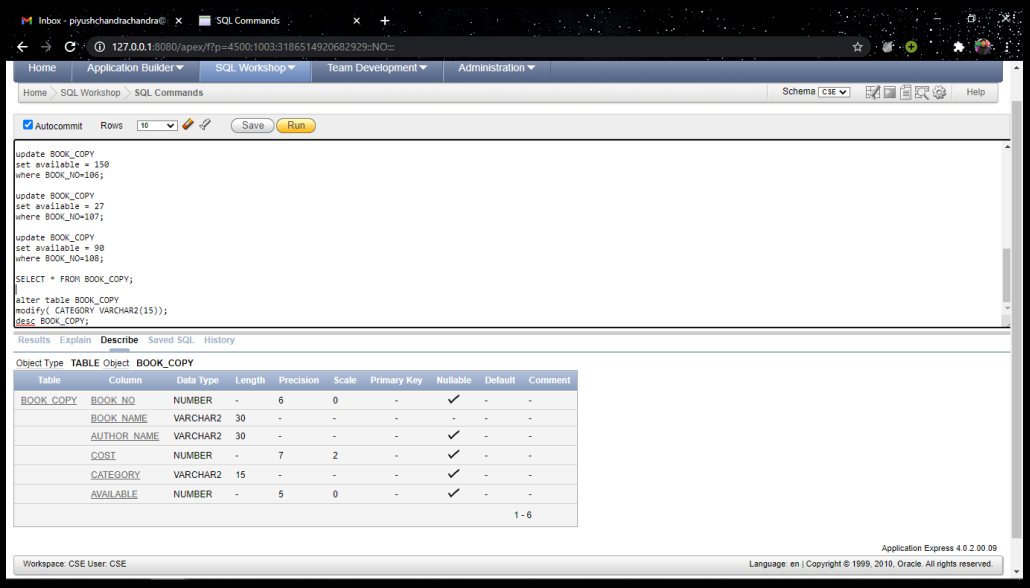

having count(BOOK_NO)>1;
insert into QUERY2 (BOOK_NAME,AUTHOR_NAME,COST)
select BOOK_NAME,AUTHOR_NAME,COST
from BOOK where COST<300;
alter table BOOK_COPY
add available number(5);
update BOOK_COPY
set available = 50
where BOOK_NO=101;
update BOOK_COPY
set available = 100
where BOOK_NO=102;
update BOOK_COPY
Table altered.
0.63 seconds


having count(BOOK_NO)>1;
insert into QUERY2 (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
Table altered.
0.63 seconds


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;
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

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;



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 SUCCEDED
  GRANT DBA TO U1;
  >GRANT SUCCEDED
  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),

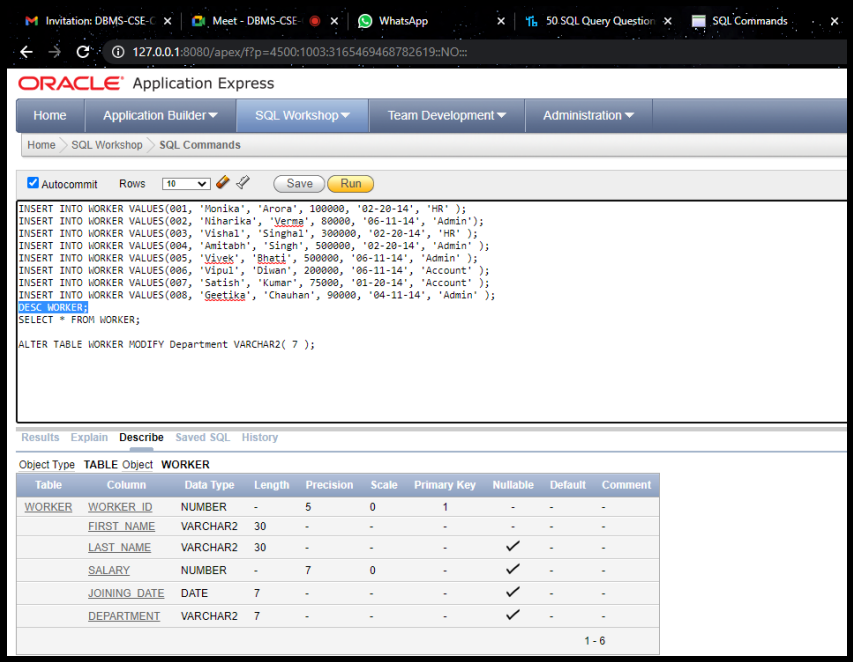
Salary number(7),

Joining_Date date,

Department varchar2(6)

);

ALTER TABLE WORKER MODIFY Department VARCHAR2(7);



(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

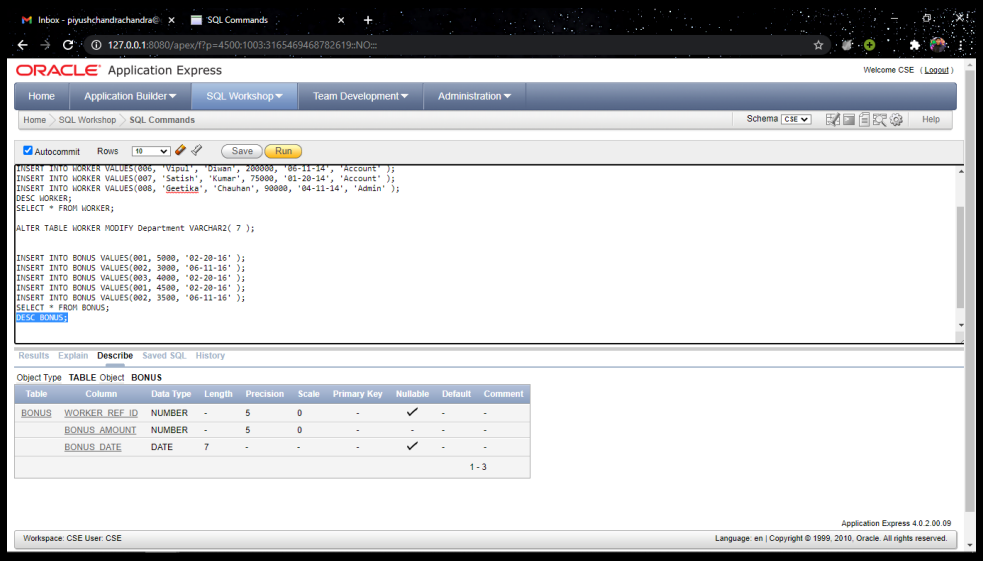
- CREATE TABLE BONUS (

  WORKER_REF_ID number(5),

  BONUS_AMOUNT number(5) NOT NULL,

  BONUS_DATE date

  );



(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 (

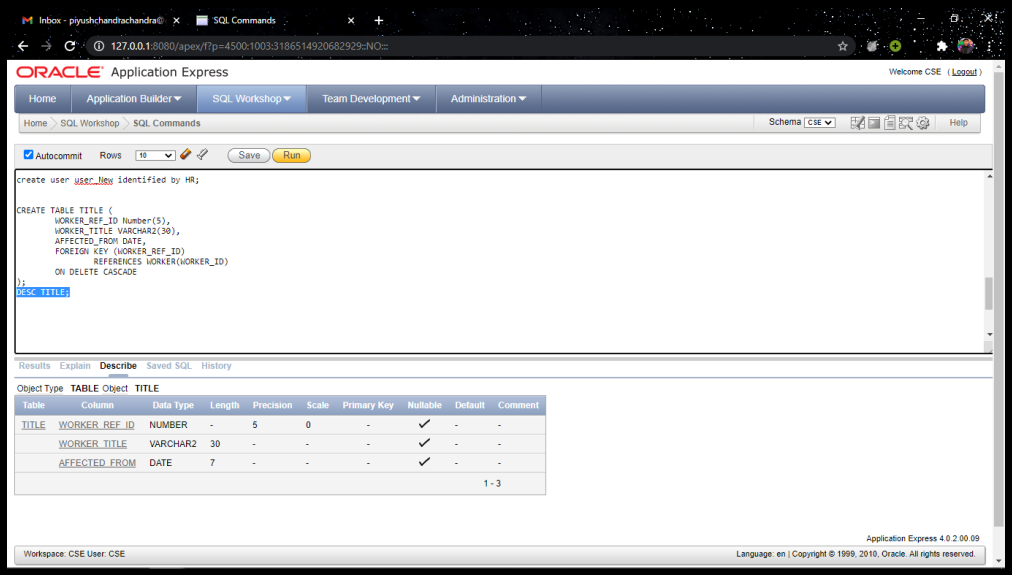
  WORKER_REF_ID Number(5),

  WORKER_TITLE VARCHAR2(30),

  AFFECTED_FROM DATE,

  FOREIGN KEY (WORKER_REF_ID)

```
        REFERENCES WORKER(WORKER_ID)
    ON DELETE CASCADE
);
```



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');

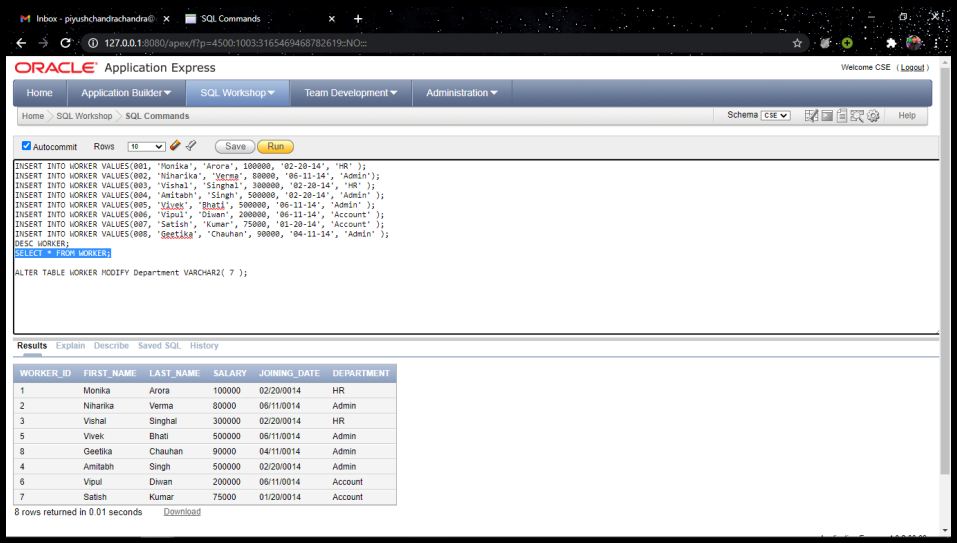
  INSERT INTO WORKER VALUES(006, 'Vipul', 'Diwan', 200000, '06-11-14', 'Account');

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;



(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');

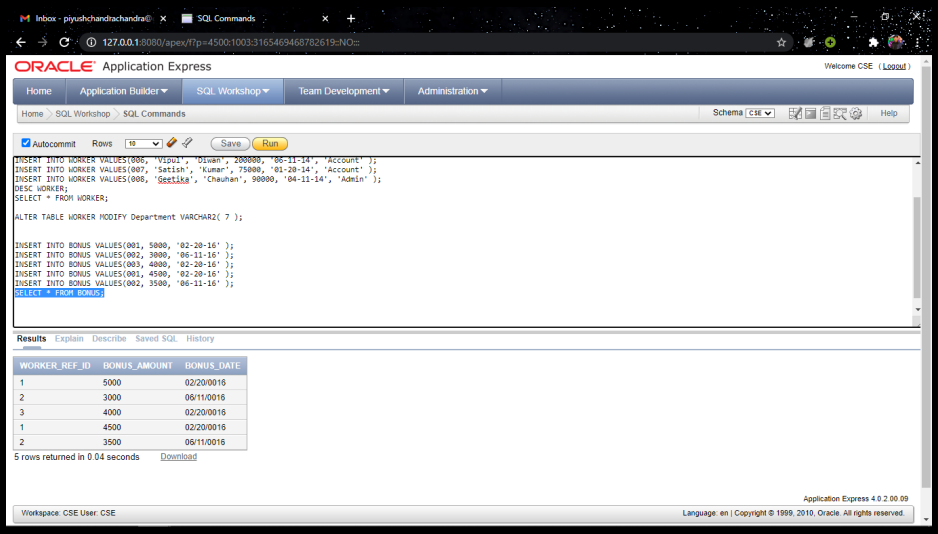
  INSERT INTO BONUS VALUES(002, 3000, '06-11-16');

  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;



(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');

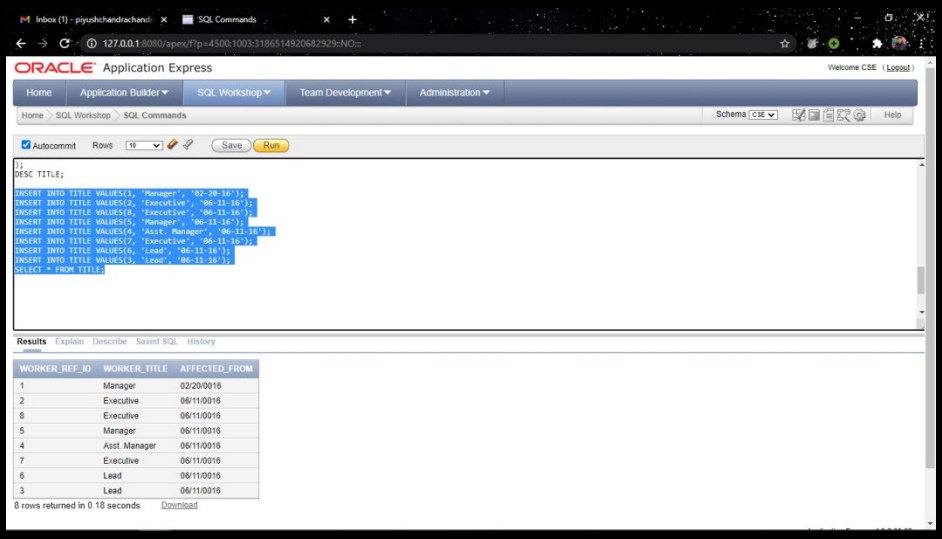
  INSERT INTO TITLE VALUES(4, 'Asst. Manager', '06-11-16');

  INSERT INTO TITLE VALUES(7, 'Executive', '06-11-16');

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

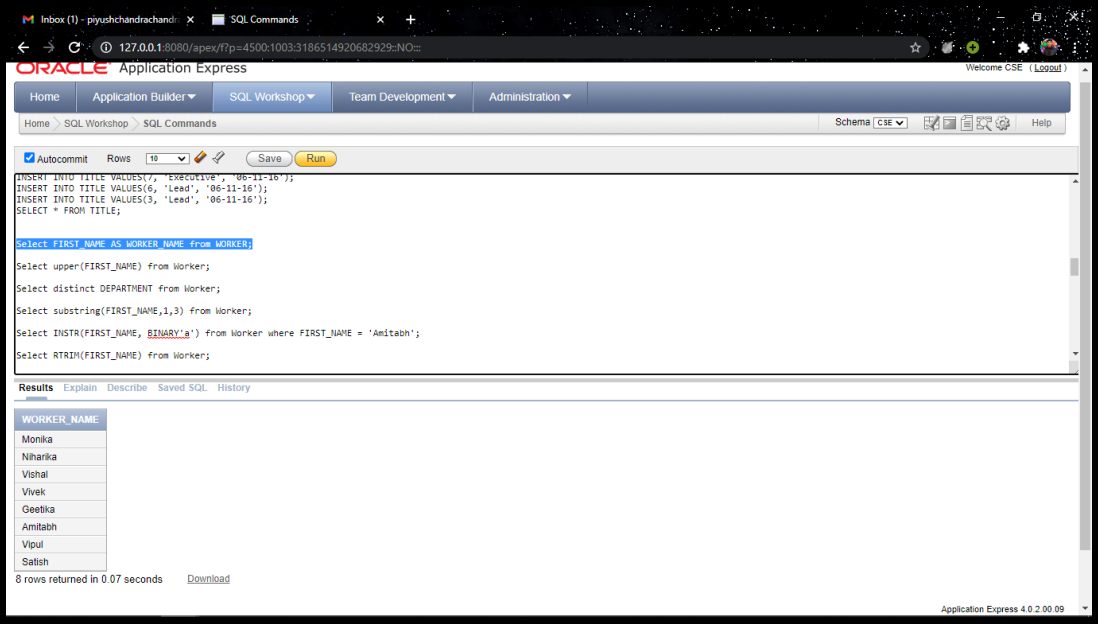
INSERT INTO TITLE VALUES(3, 'Lead', '06-11-16');

SELECT * FROM TITLE;



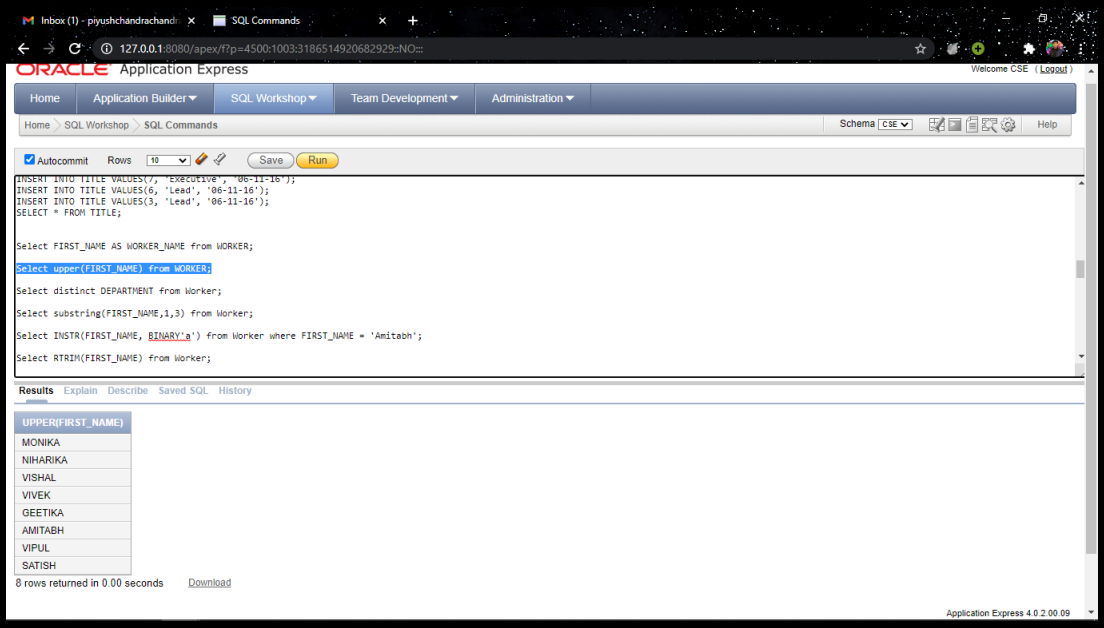
1. Write An SQL Query To Fetch "FIRST_NAME" From Worker Table Using The Alias Name WORKER_NAME.

➢ Select FIRST_NAME AS WORKER_NAME from WORKER;



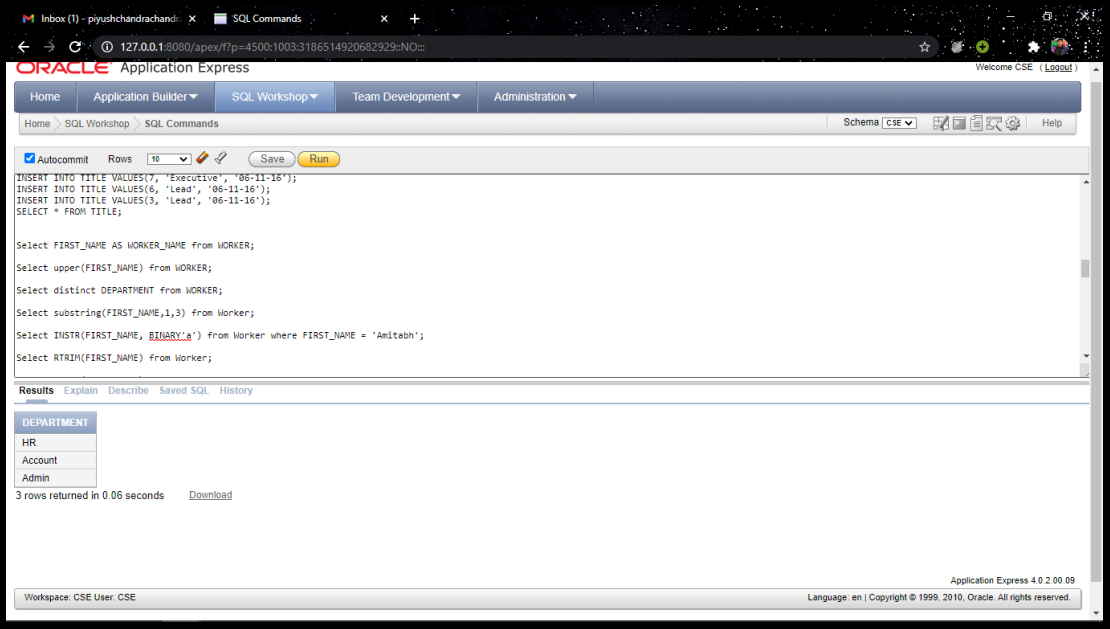
2. Write An SQL Query To Fetch “FIRST_NAME” From Worker Table In Upper Case.

➢ Select upper(FIRST_NAME) from WORKER;



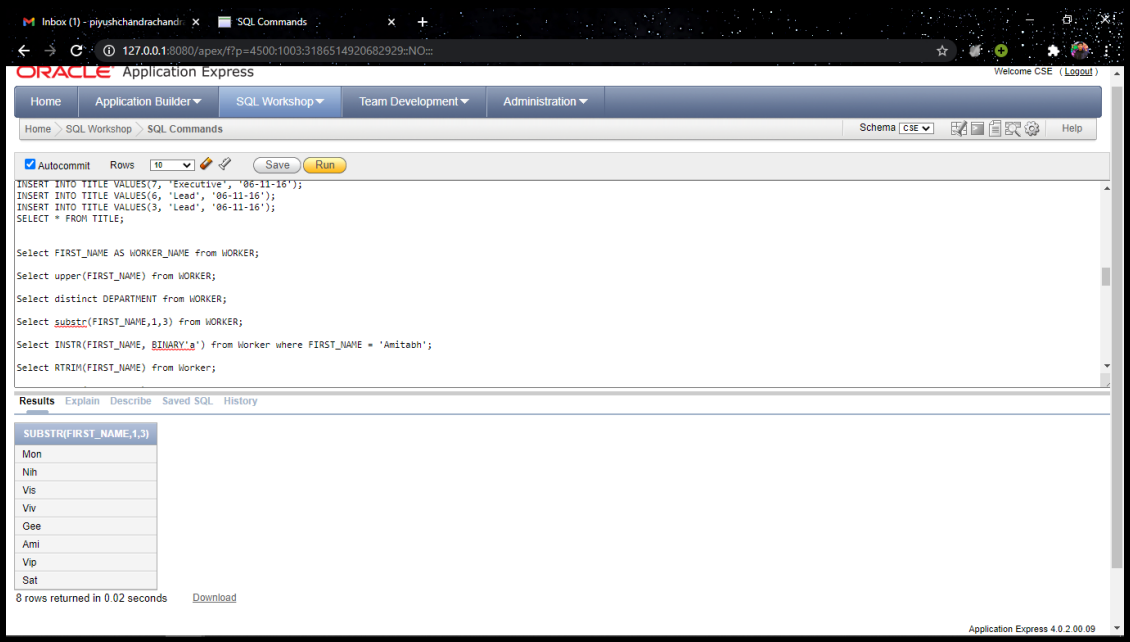
3. Write An SQL Query To Fetch Unique Values Of DEPARTMENT From Worker Table.

➢ Select distinct DEPARTMENT from WORKER;



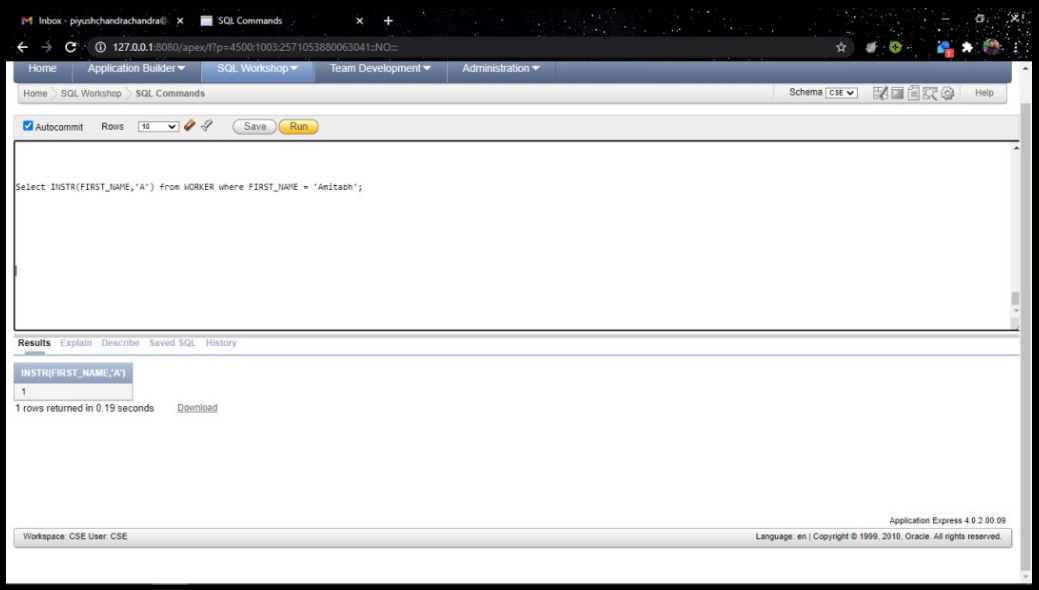
4. Write an SQL Query to Print the First Three Characters Of FIRST_NAME from Worker Table.

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



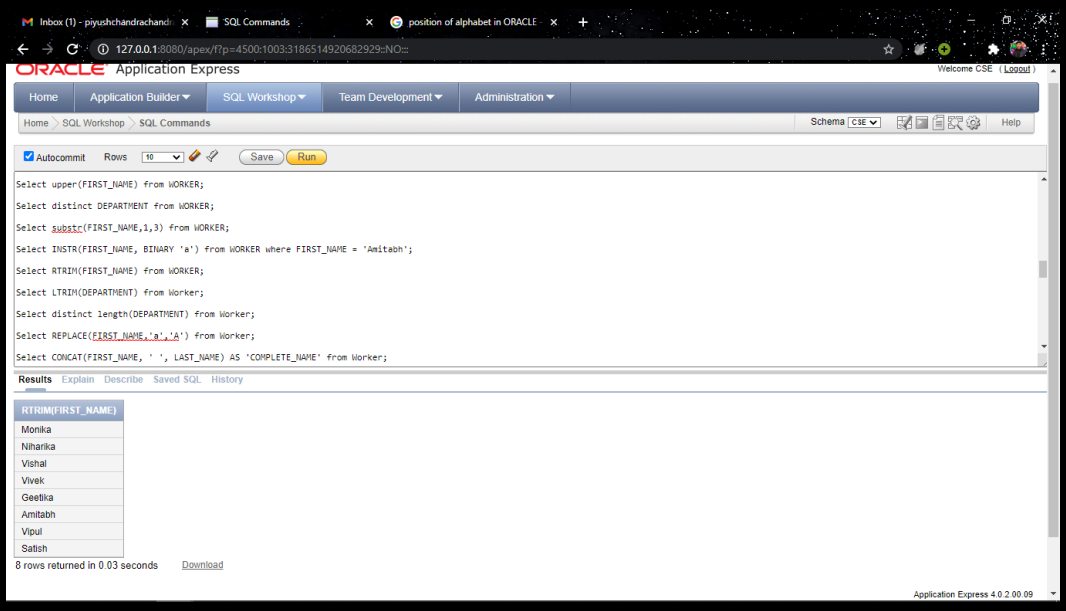
5. Write an SQL Query To Find The Position Of The Alphabet („A") In The First Name Column „Amitabh" From Worker Table.

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



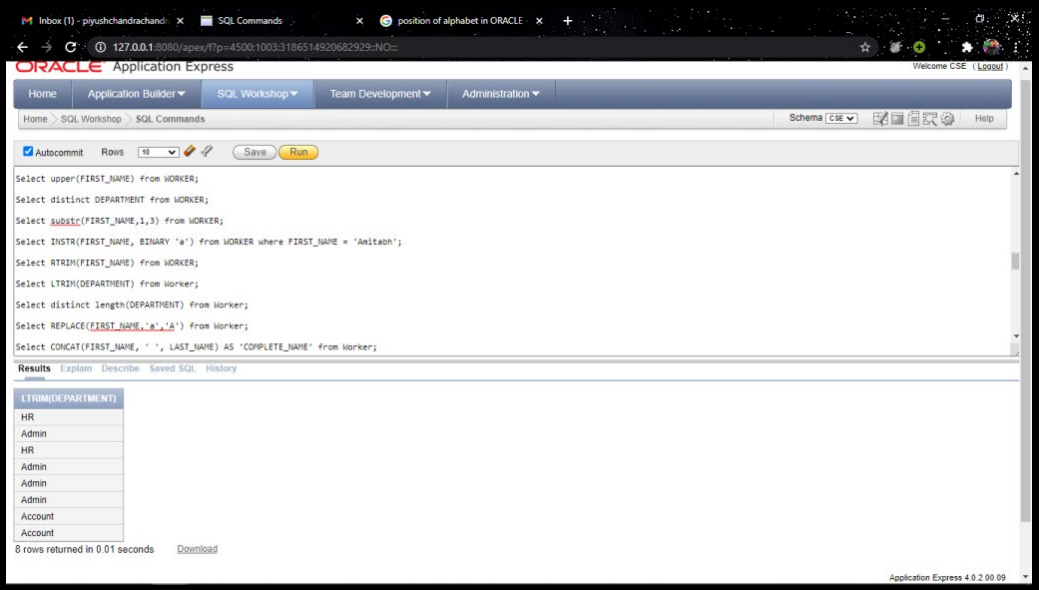
6. Write An SQL Query To Print The FIRST_NAME From Worker Table After Removing White Spaces From The Right Side.

➢ Select RTRIM(FIRST_NAME) from WORKER;



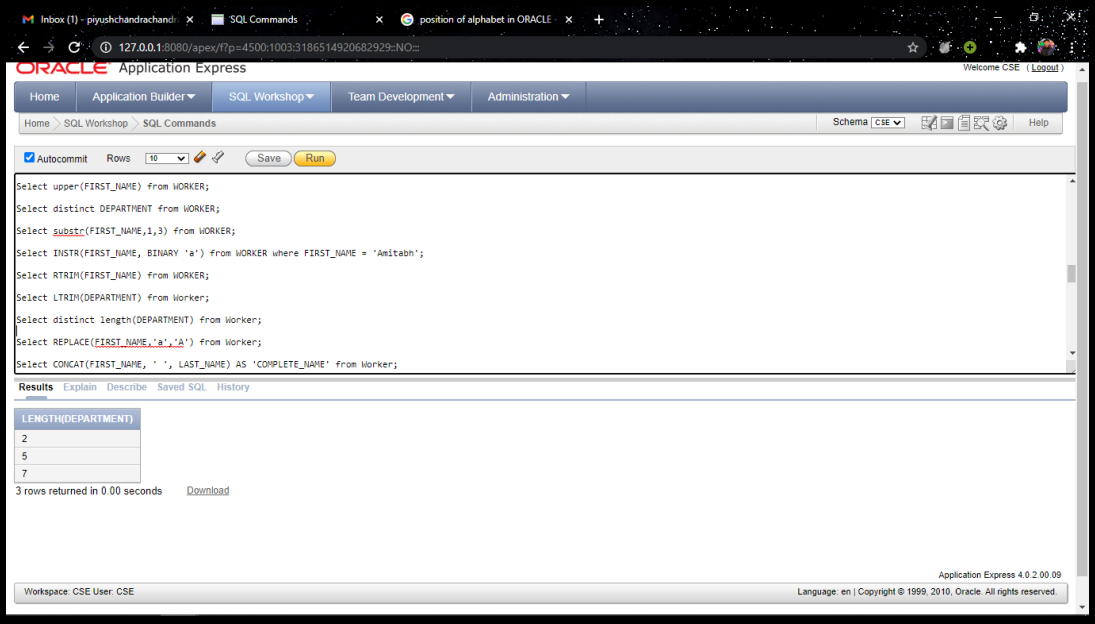
7. Write An SQL Query To Print The DEPARTMENT From Worker Table After Removing White Spaces From The Left Side.

➢ Select LTRIM(DEPARTMENT) from Worker;



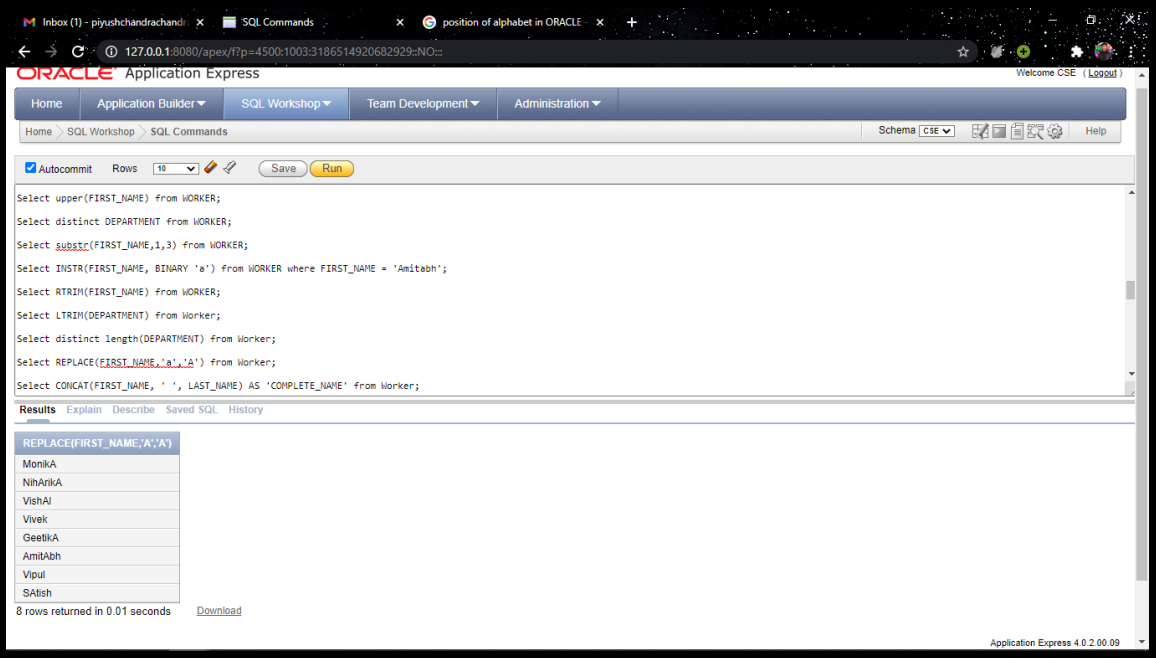
8. Write An SQL Query That Fetches The Unique Values Of DEPARTMENT From Worker Table And Prints Its Length.

➢ Select distinct length(DEPARTMENT) from Worker;



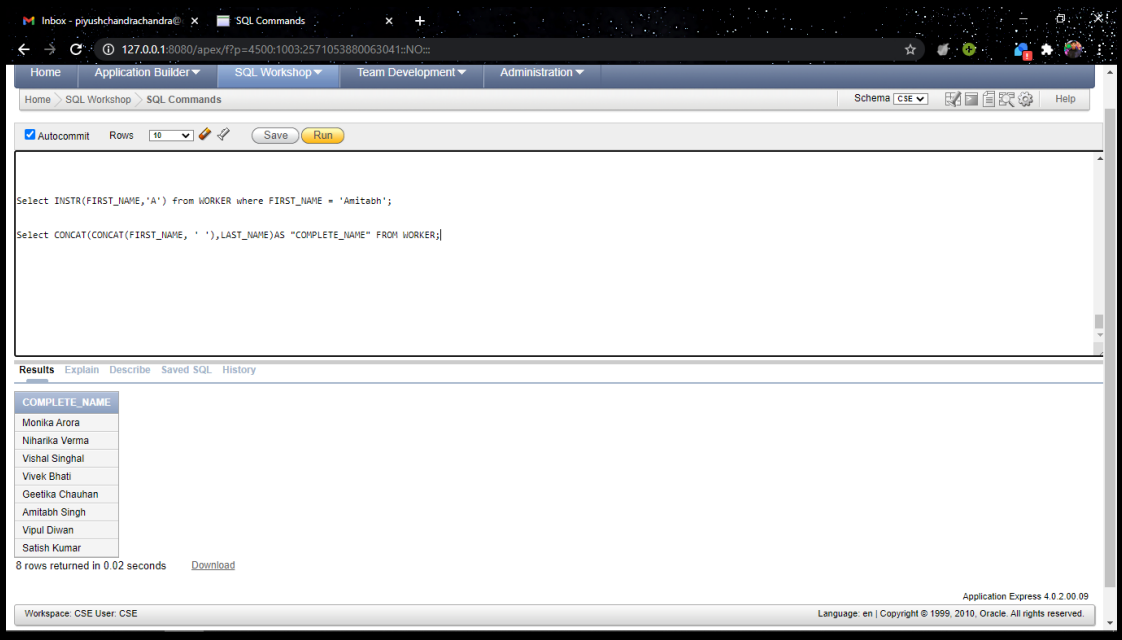
9. Write An SQL Query To Print The FIRST_NAME From Worker Table After Replacing „A" With „A".

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



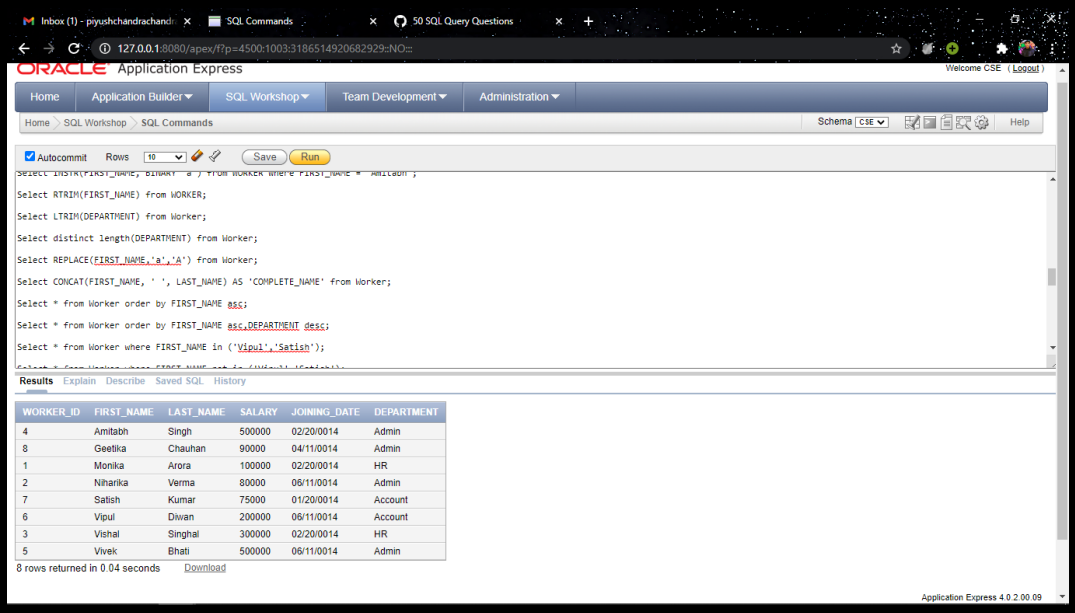
10. Write An SQL Query To Print The FIRST_NAME And LAST_NAME From Worker Table Into A Single Column COMPLETE_NAME. A Space Char Should Separate Them.

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



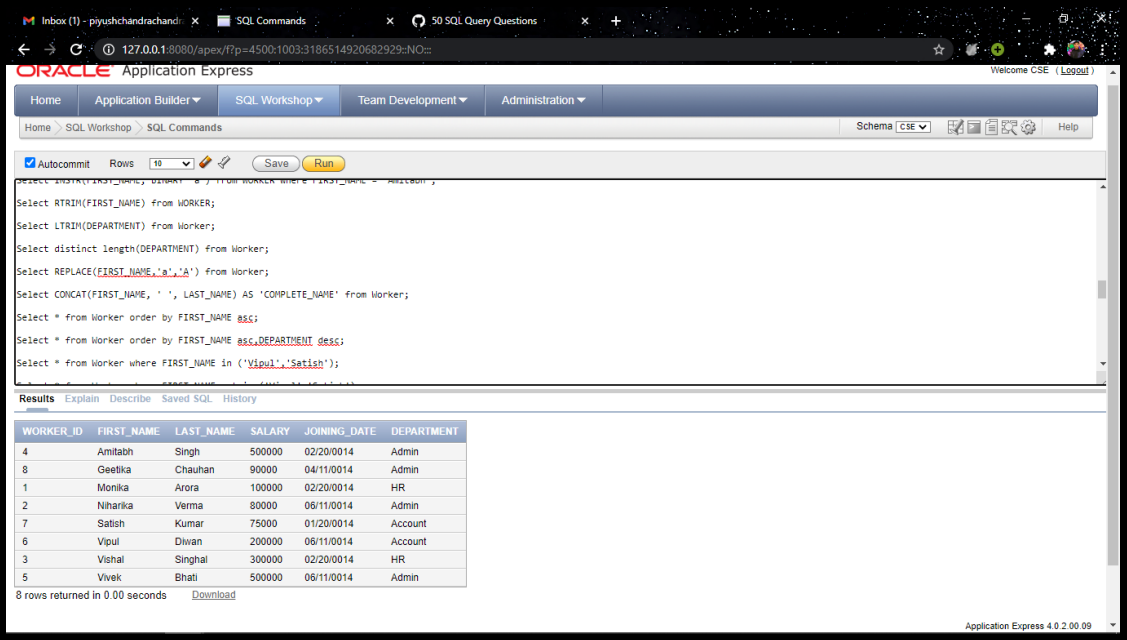
11. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST_NAME Ascending.

➢ Select * from Worker order by FIRST_NAME asc;



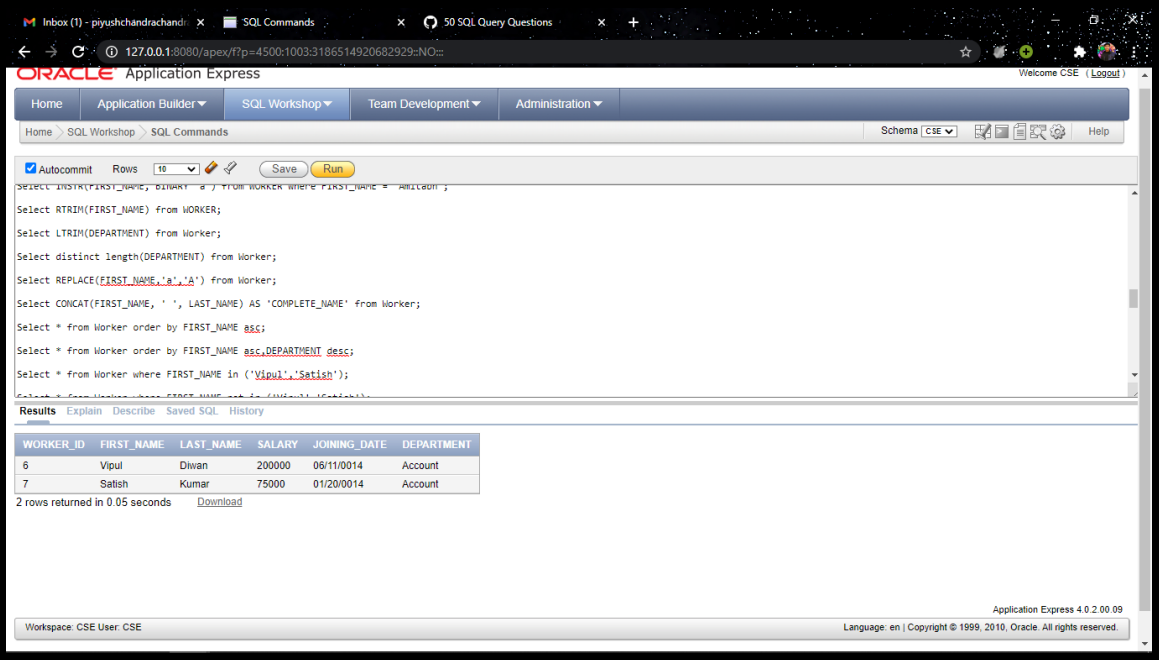
12. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST_NAME Ascending And DEPARTMENT Descending.

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



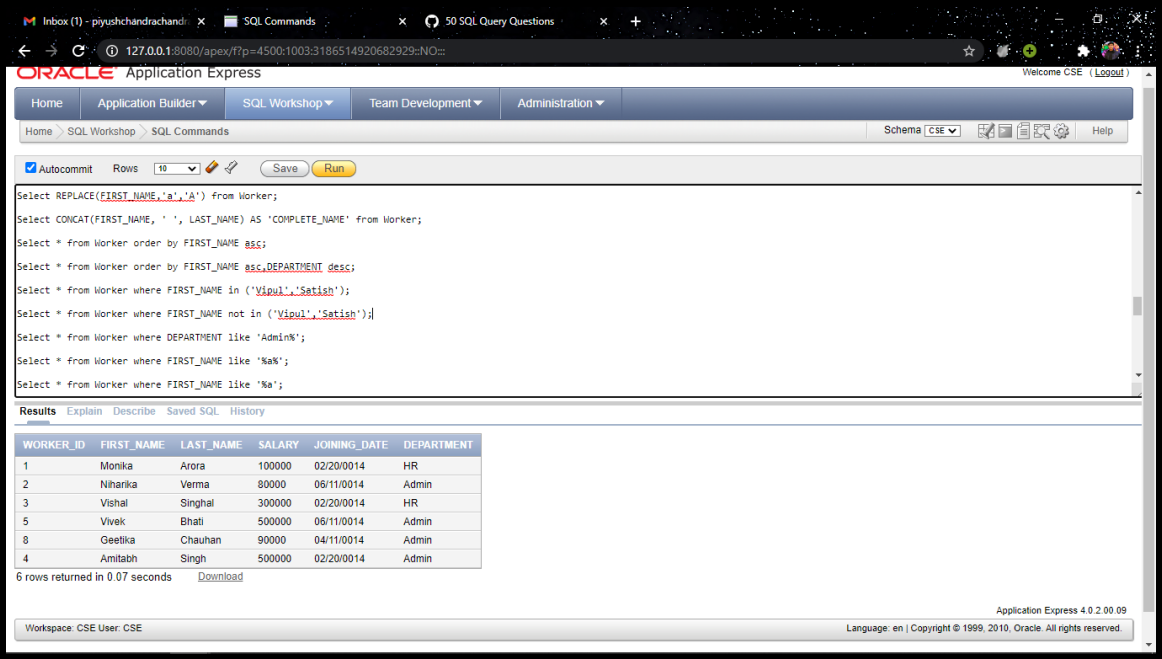
13. Write An SQL Query To Print Details For Workers With The First Name As “Vipul” And “Satish” From Worker Table.

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



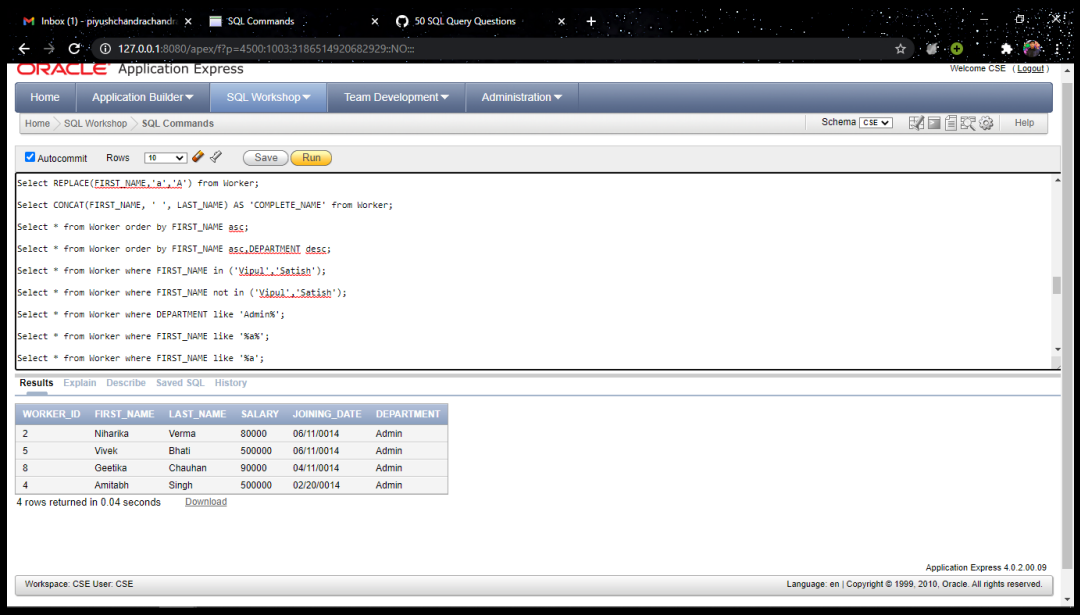
14. Write An SQL Query To Print Details Of Workers Excluding First Names, “Vipul” And “Satish” From Worker Table.

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



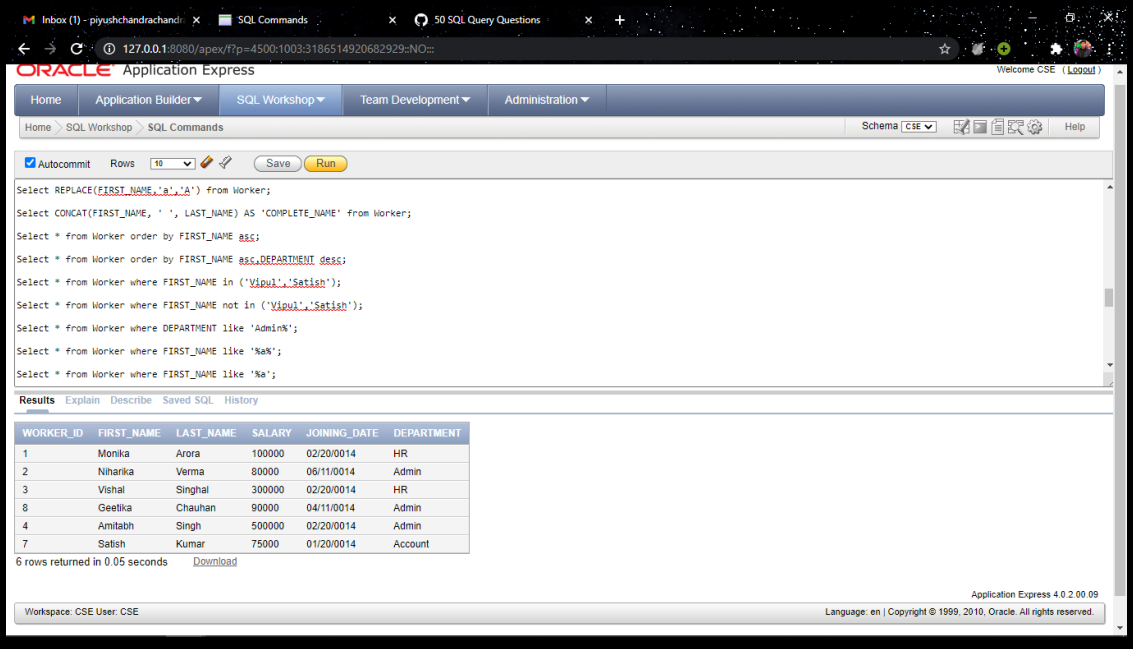
15. Write An SQL Query To Print Details Of Workers With DEPARTMENT Name As “Admin”.

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



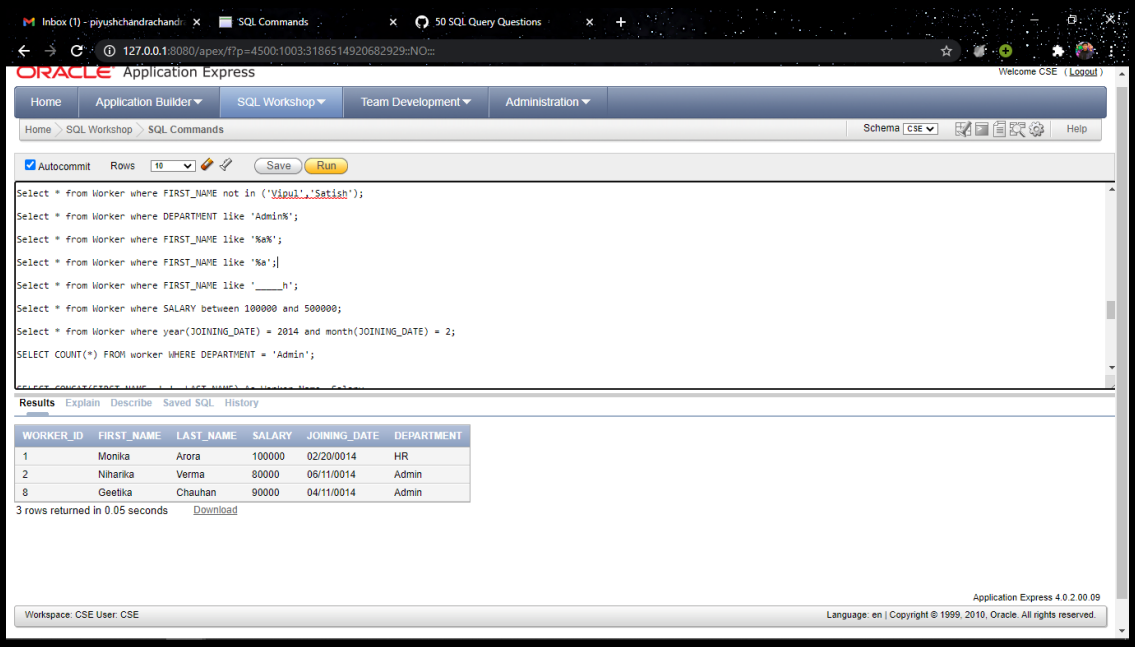
16. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Contains „A".

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



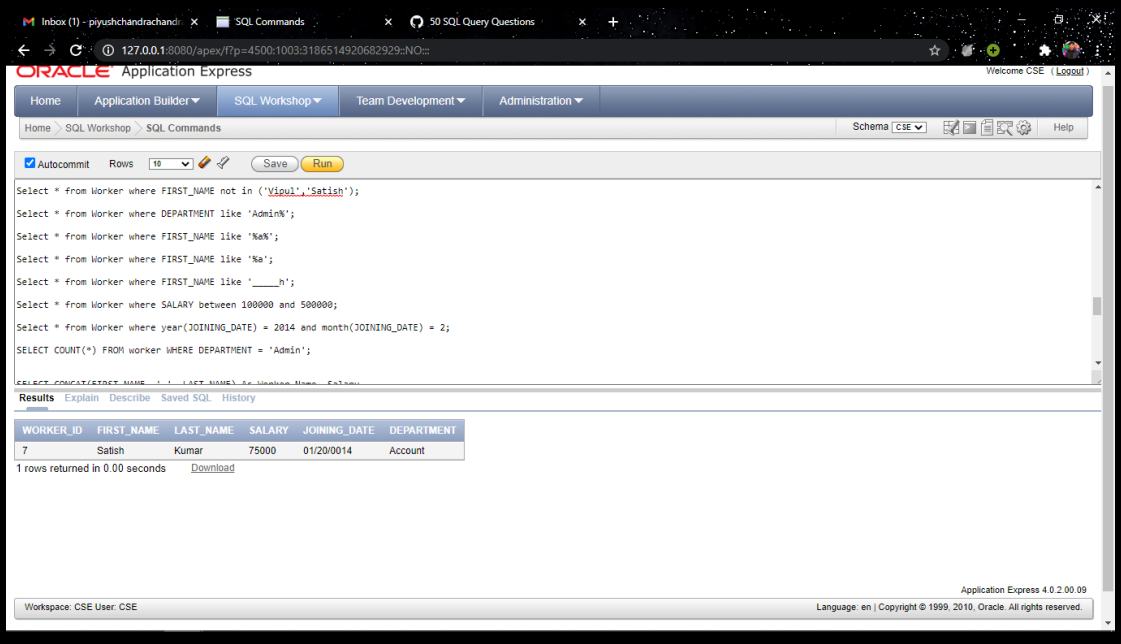
17. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Ends With „A".

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



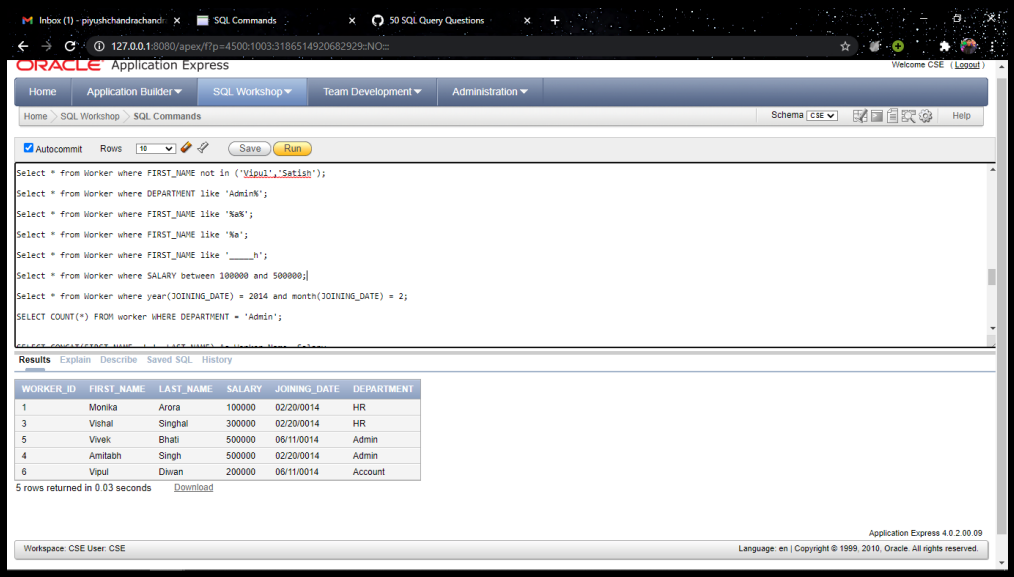
18. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Ends With „H" And Contains Six Alphabets.

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



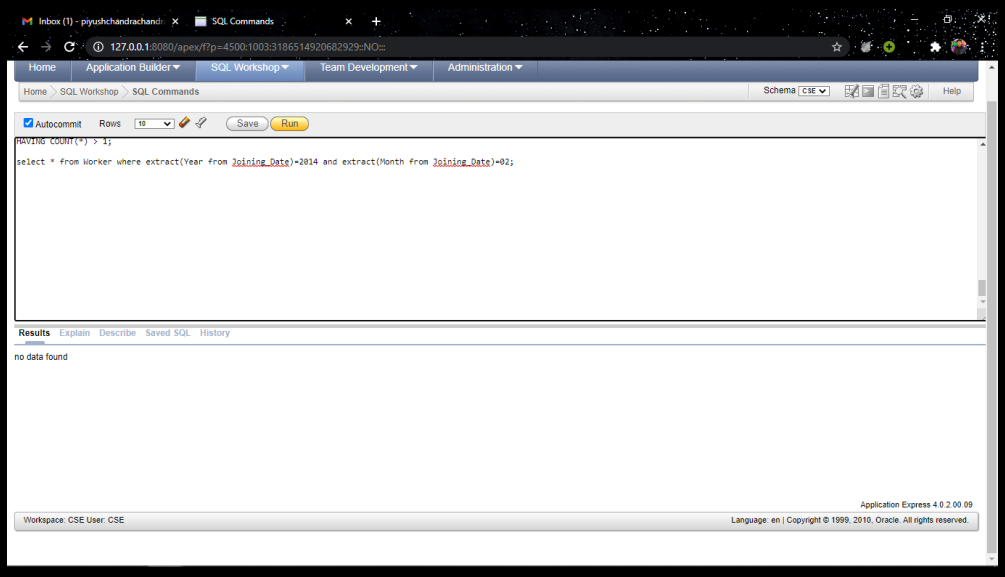
19. Write An SQL Query To Print Details Of The Workers Whose SALARY Lies Between 100000 And 500000

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



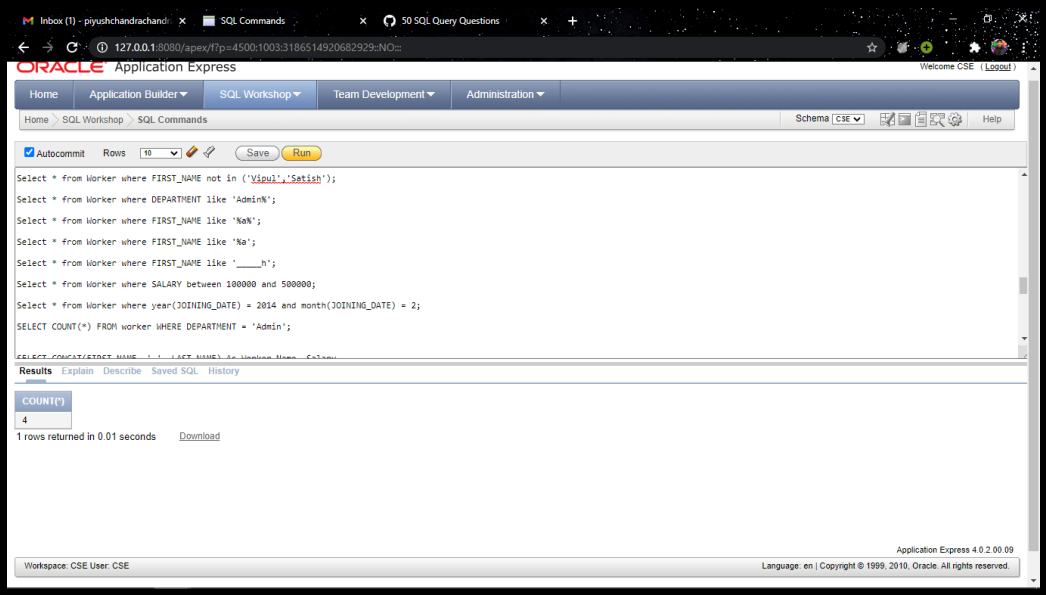
20. Write An SQL Query To Print Details Of The Workers Who Have Joined In Feb‟2014.

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



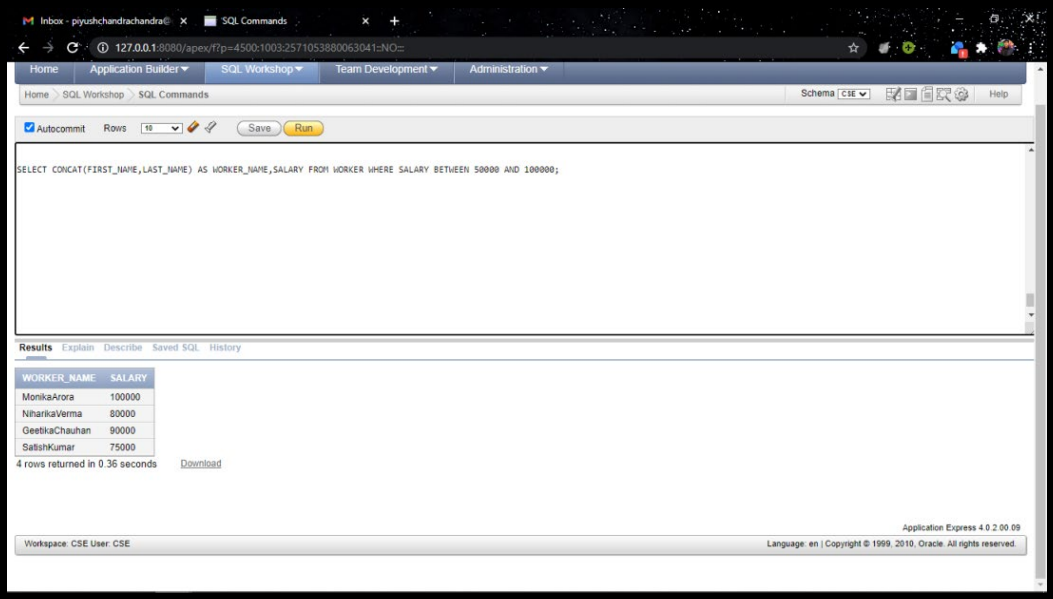
21. Write An SQL Query To Fetch The Count Of Employees Working In The Department „Admin‟

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



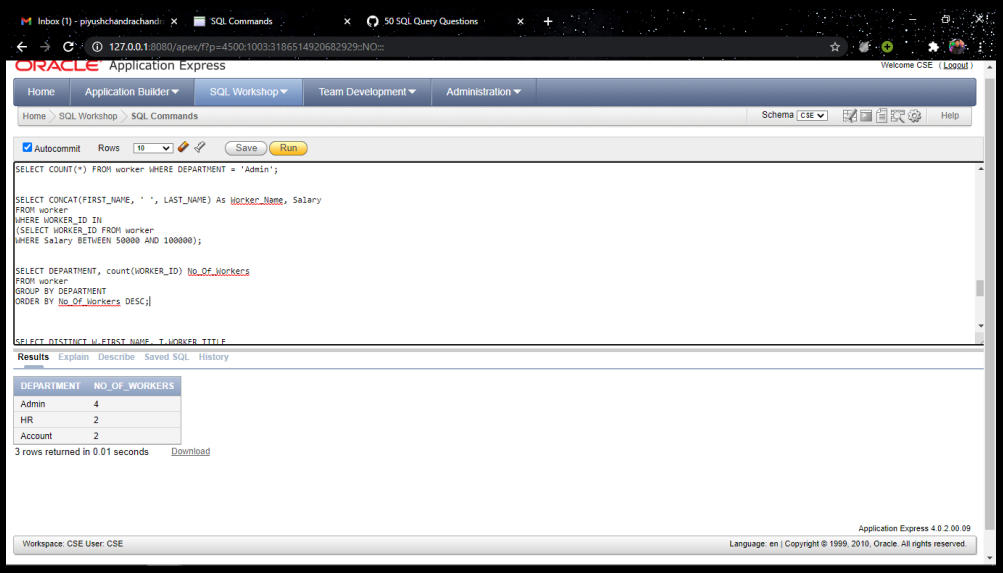
22. Write An SQL Query To Fetch Worker Names With Salaries >= 50000 And <= 100000.

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



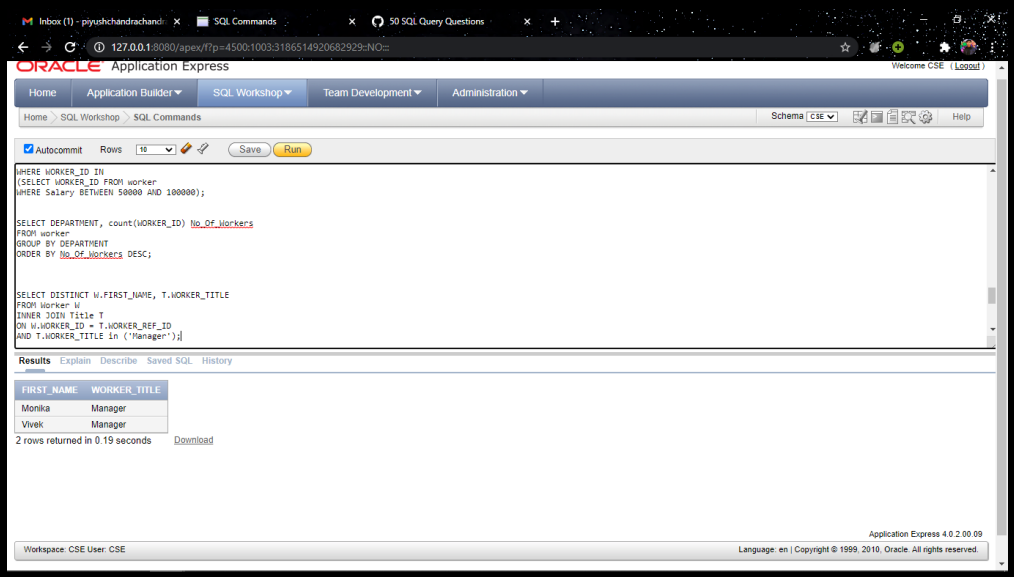
23. Write An SQL Query To Fetch The No. Of Workers for Each Department In The Descending Order.

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



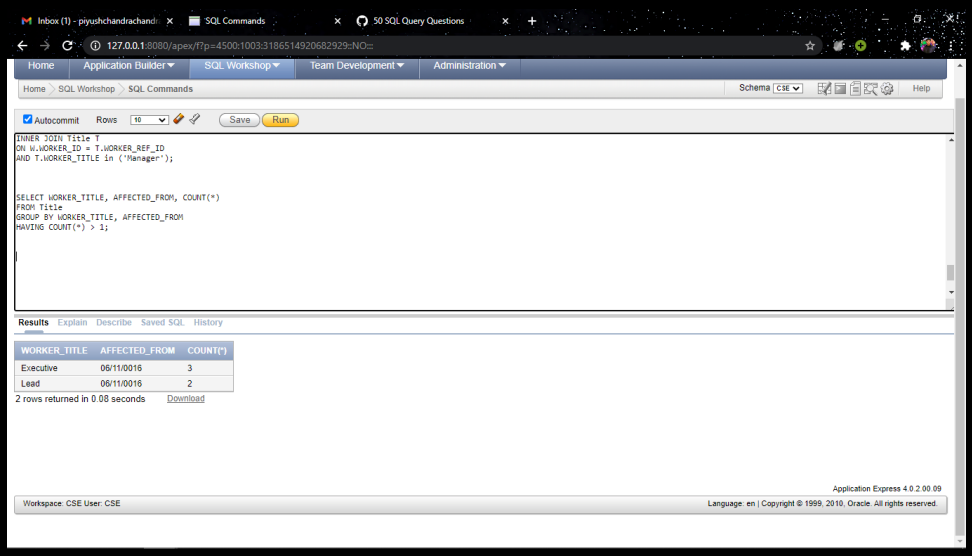
24. Write An SQL Query To Print Details Of The Workers Who Are Also Managers.

➢
```
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');
```



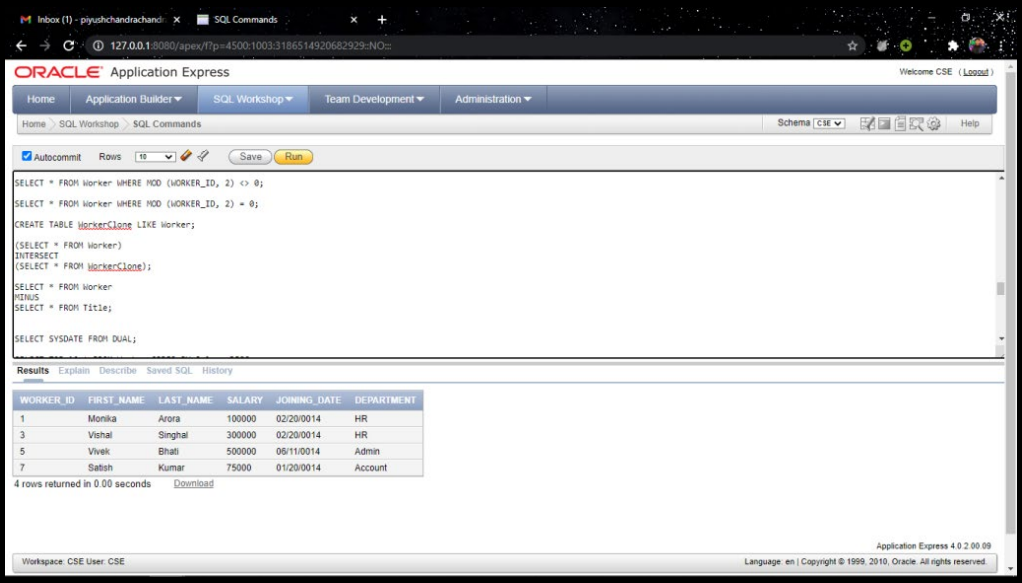
25. Write An SQL Query To Fetch Duplicate Records Having Matching Data In Some Fields Of A Table.

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



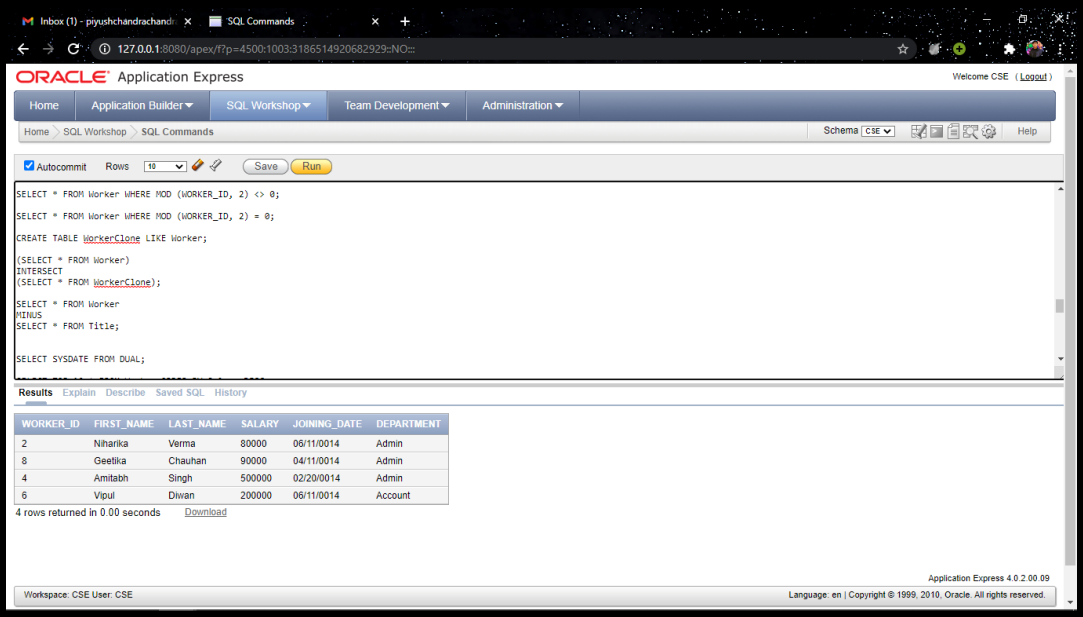
26. Write An SQL Query To Show Only Odd Rows From worker Table.

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



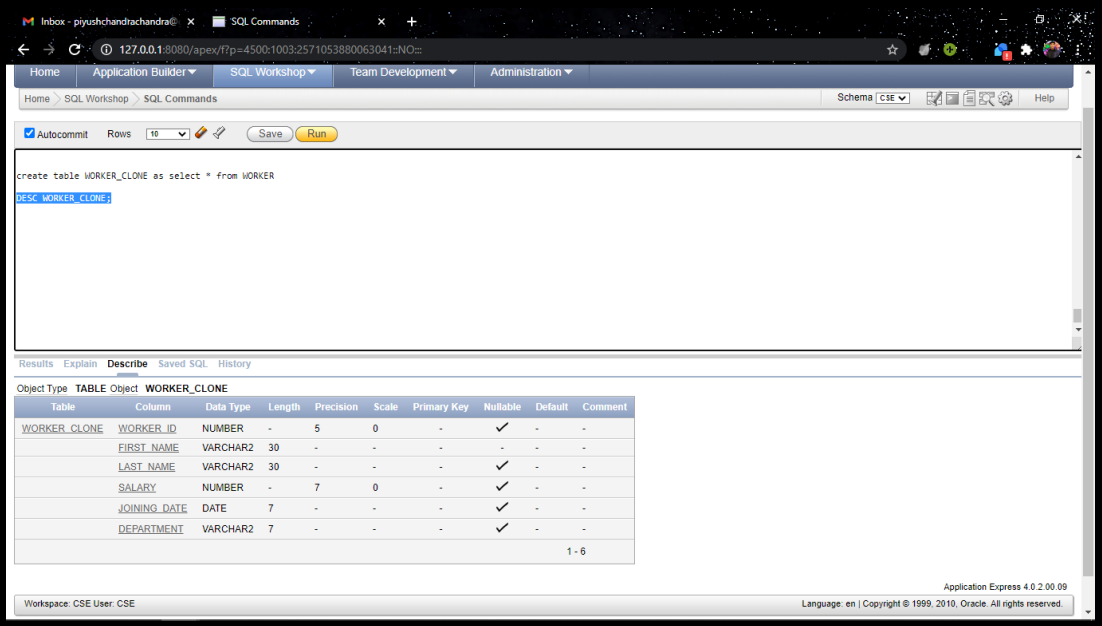
27. Write An SQL Query To Show Only Even Rows From worker Table.

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



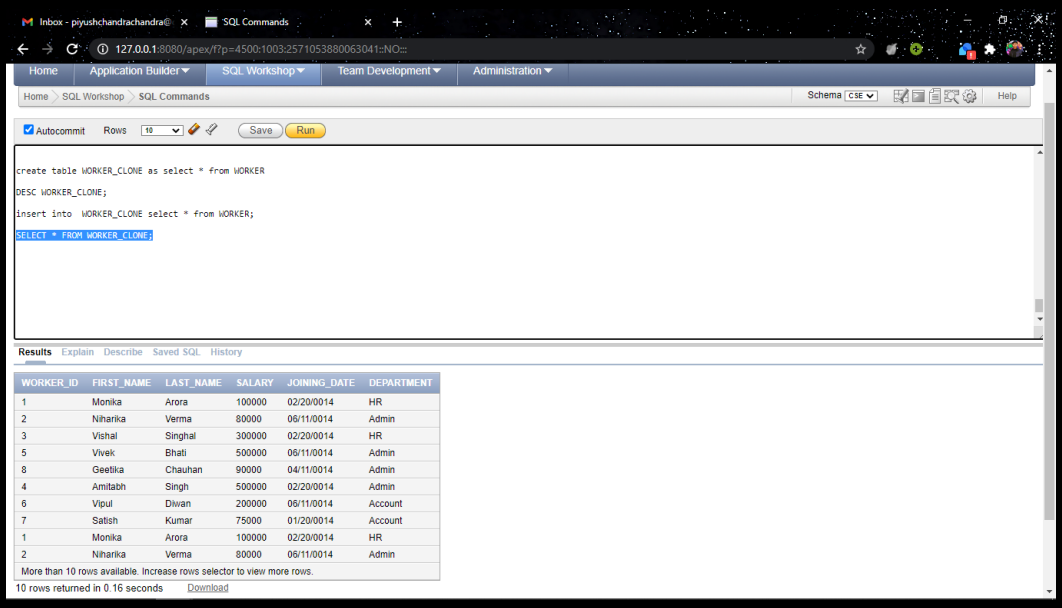
28. Write An SQL Query To Clone New_Worker Table From Worker Table.

- create table WORKER_CLONE as select * from WORKER



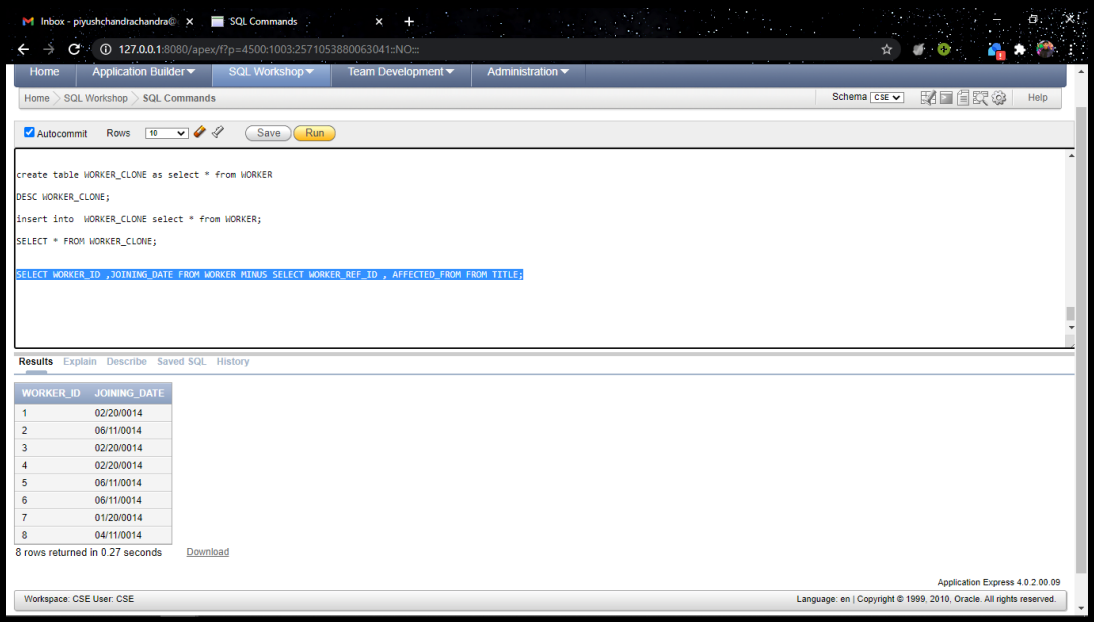
29. Write An SQL Query To Fetch Intersecting Records Of Worker and New_Worker Tables.

- insert into WORKER_CLONE select * from WORKER;



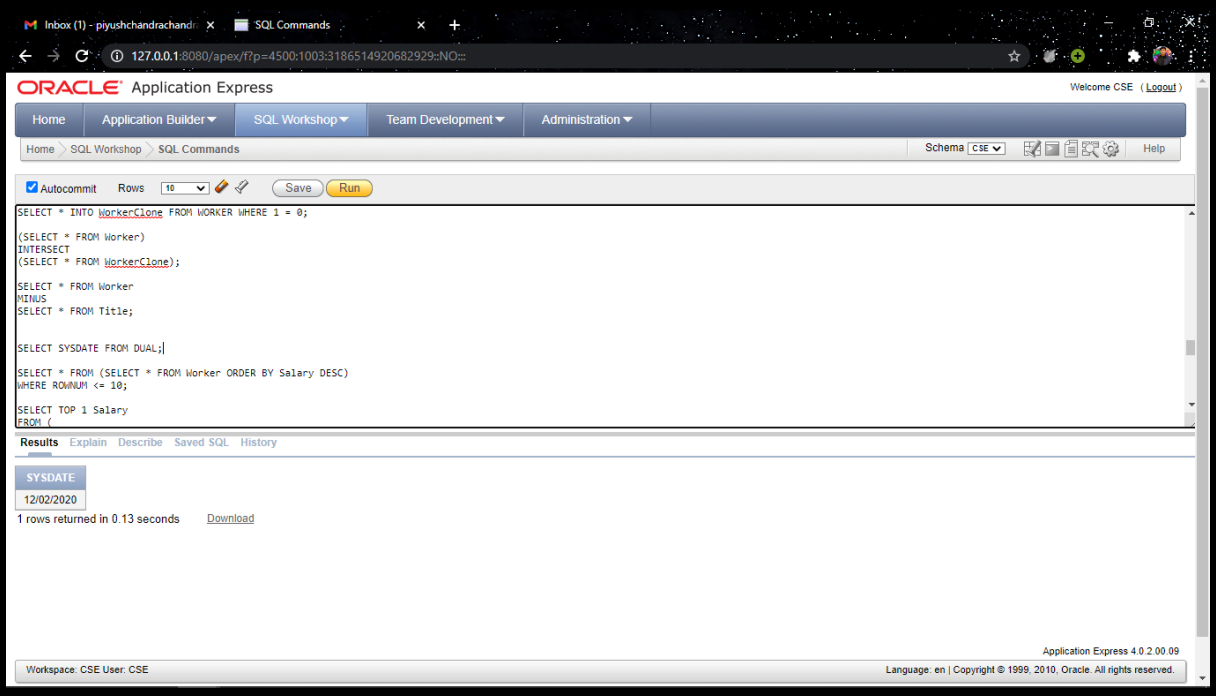
30. Write An SQL Query To Show Records From Worker table That Title Table Does Not Have.

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



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

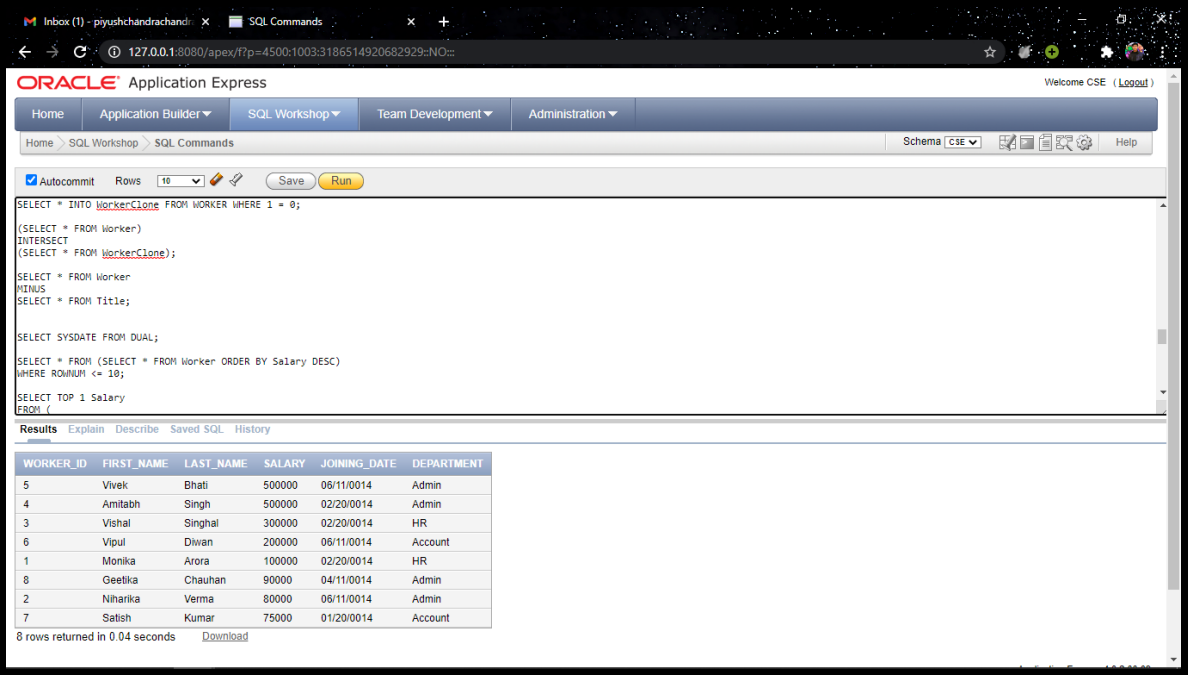
- SELECT SYSDATE FROM DUAL;



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;

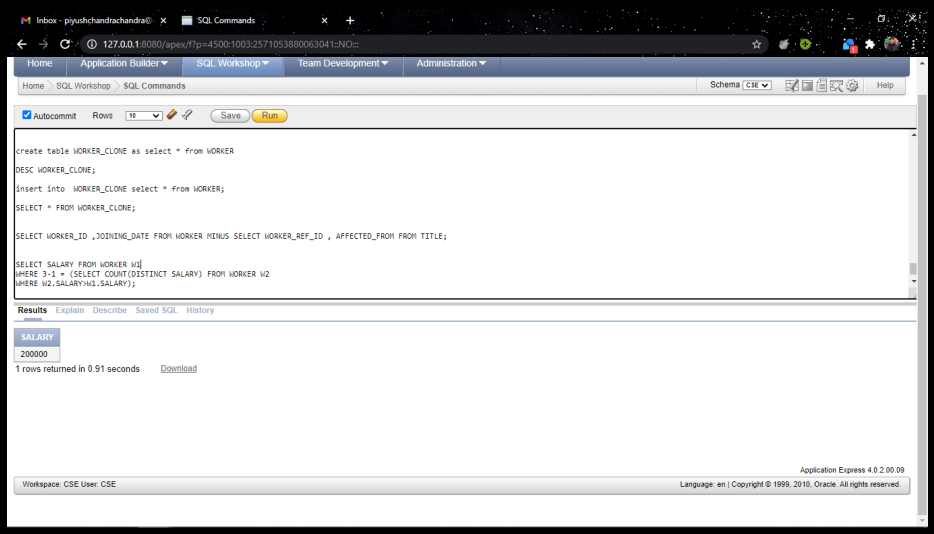


33. Write An SQL Query To Determine The Nth (Say 3) Highest Salary From Worker Table.

- SELECT SALARY FROM WORKER W1

WHERE 3-1 = (SELECT COUNT(DISTINCT SALARY) FROM WORKER W2

WHERE W2.SALARY>W1.SALARY);



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

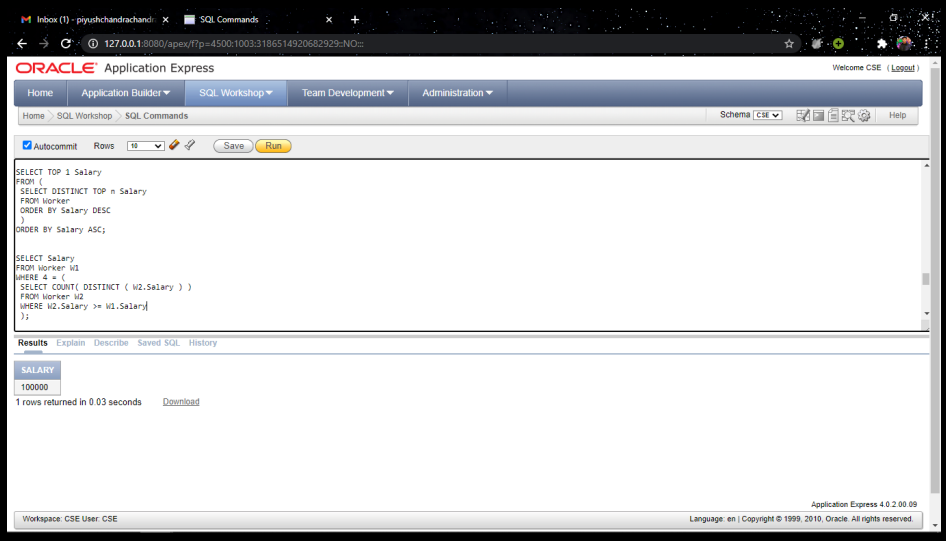
  WHERE 4 = (

  SELECT COUNT(DISTINCT (W2.Salary))

  FROM Worker W2

  WHERE W2.Salary >= W1.Salary

  );



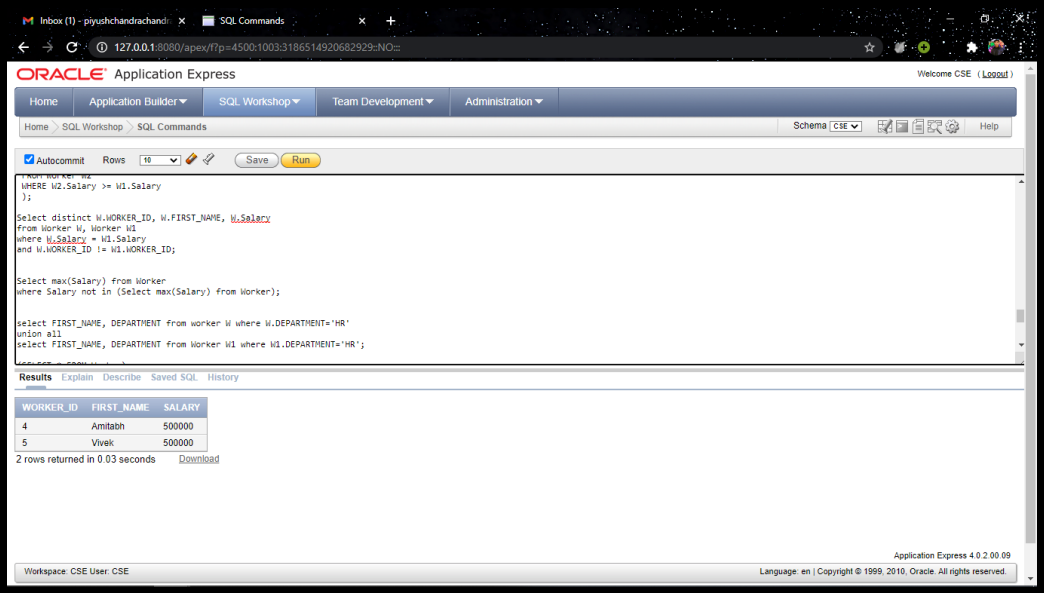
35. Write An SQL Query To Fetch The List Of Employees With The Same Salary from Worker Table.

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

from Worker W, Worker W1

where W.Salary = W1.Salary

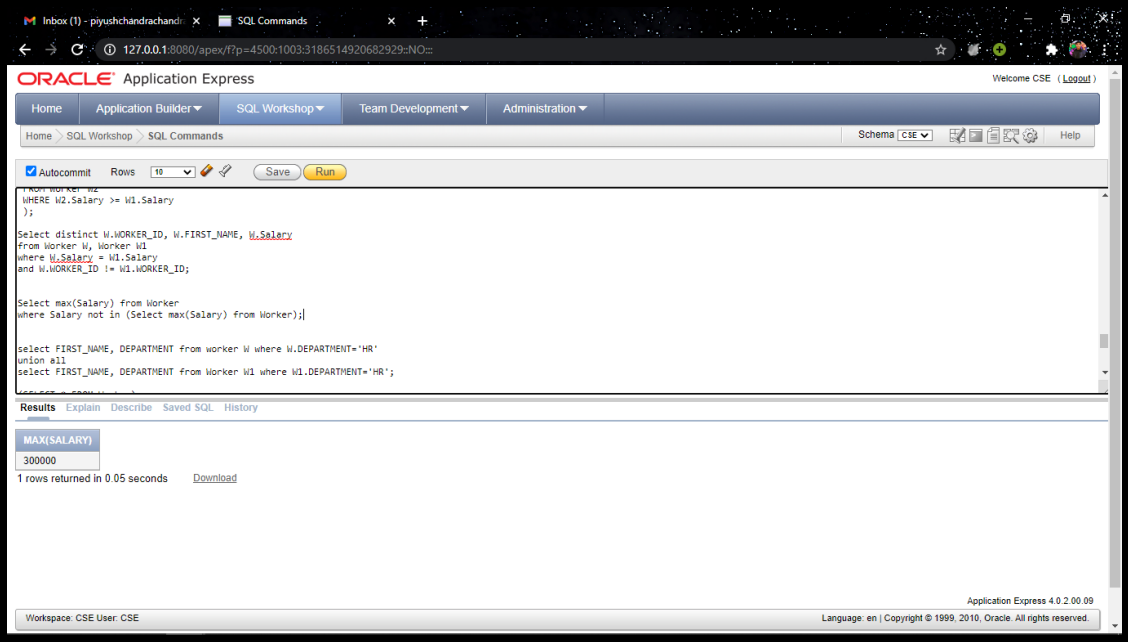
and W.WORKER_ID != W1.WORKER_ID;



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);

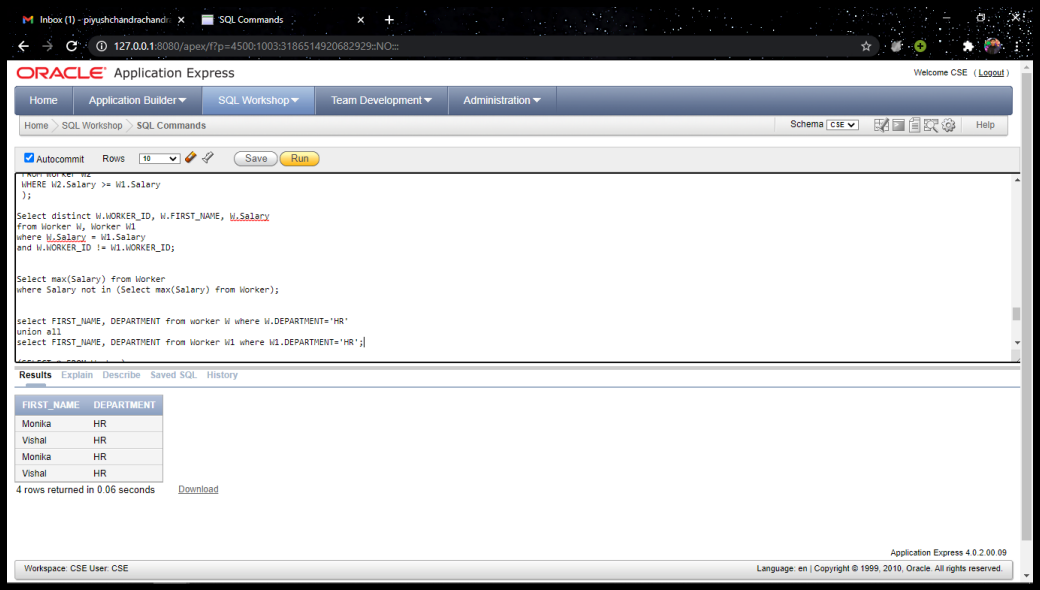


37. Write An SQL Query To Show One Row Twice In Results From Worker Table.

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

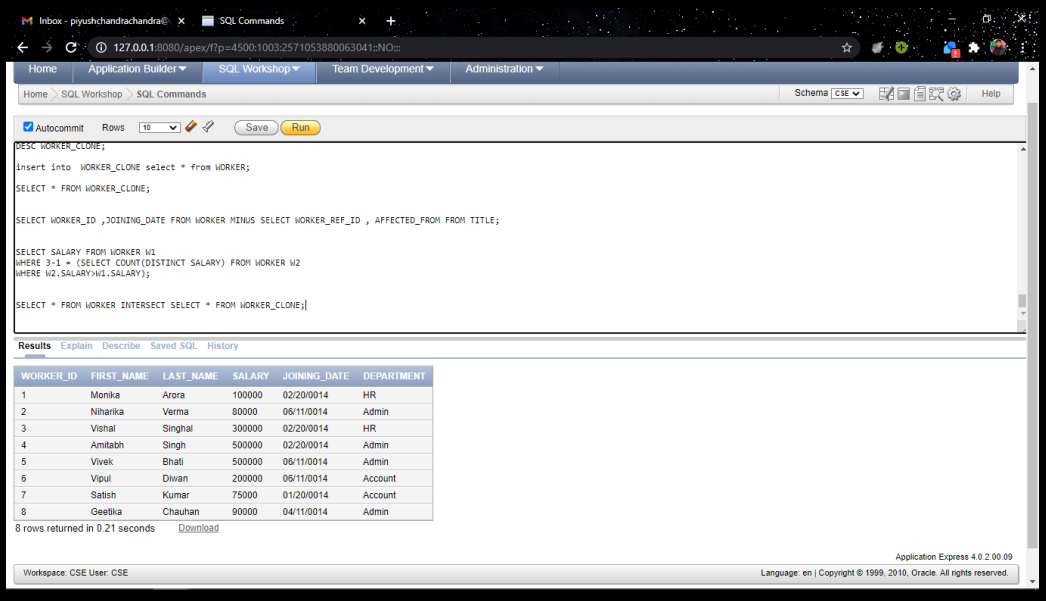
  union all

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



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

- SELECT * FROM WORKER INTERSECT SELECT * FROM WORKER_CLONE;

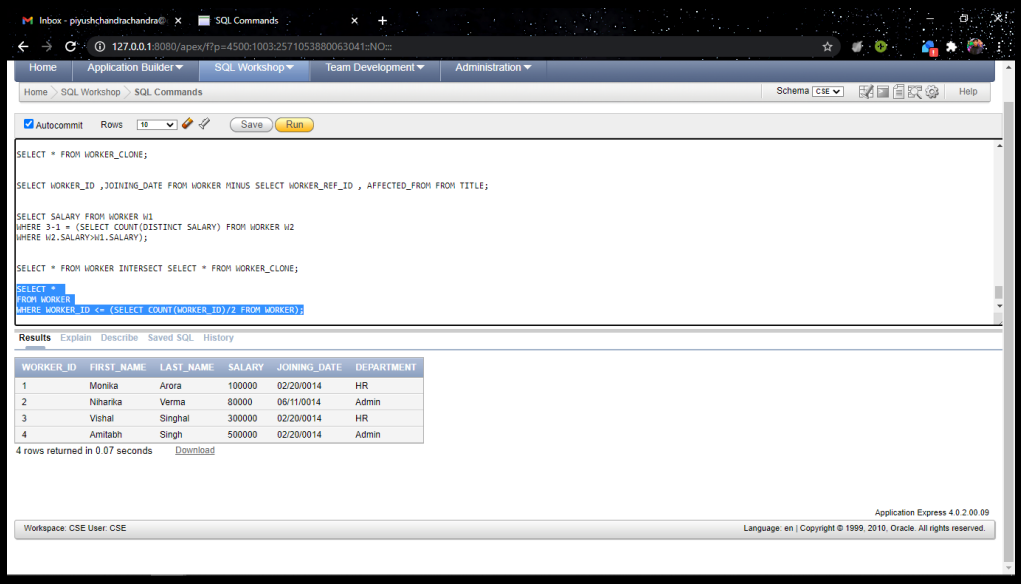


39. Write An SQL Query To Fetch The First 50% Records From Worker Table.

- SELECT *

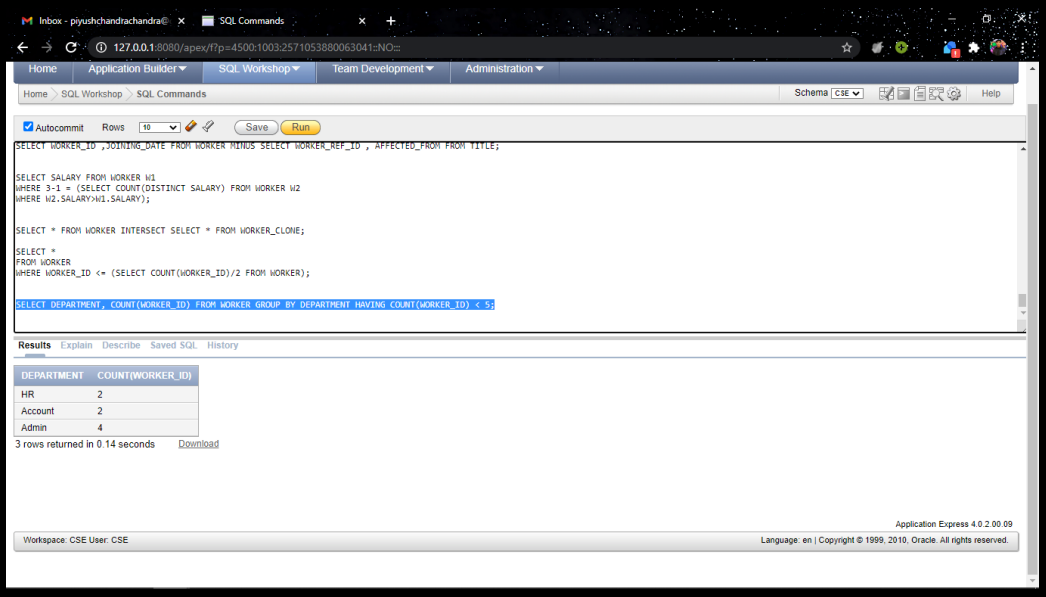
  FROM WORKER

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



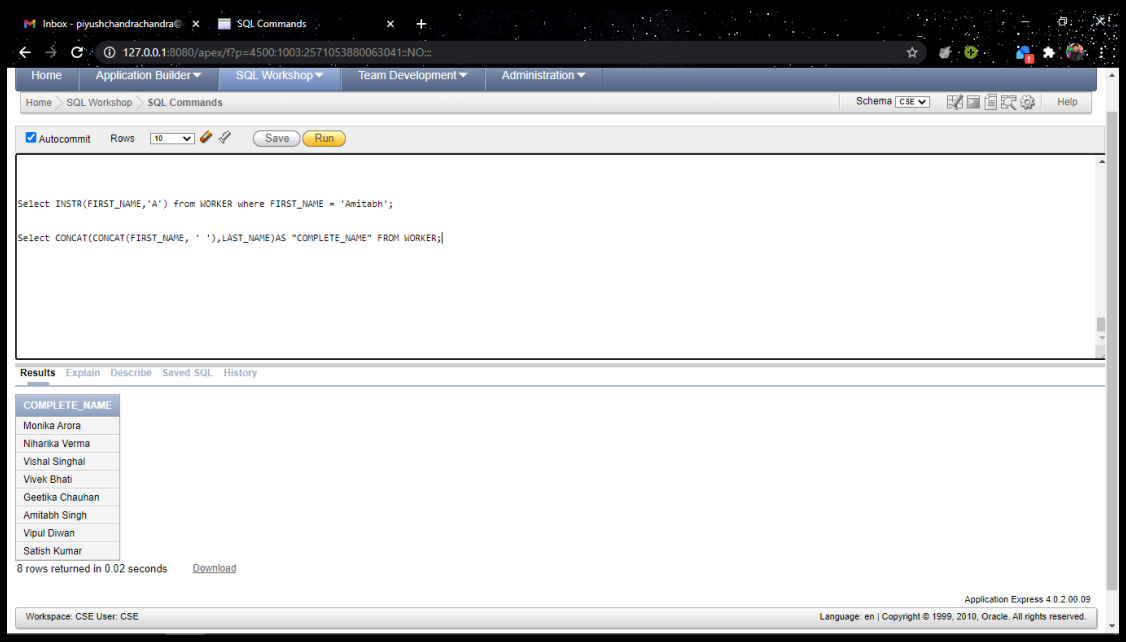
40. Write An SQL Query To Fetch The Departments That Have Less Than Five People In It from Worker Table.

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



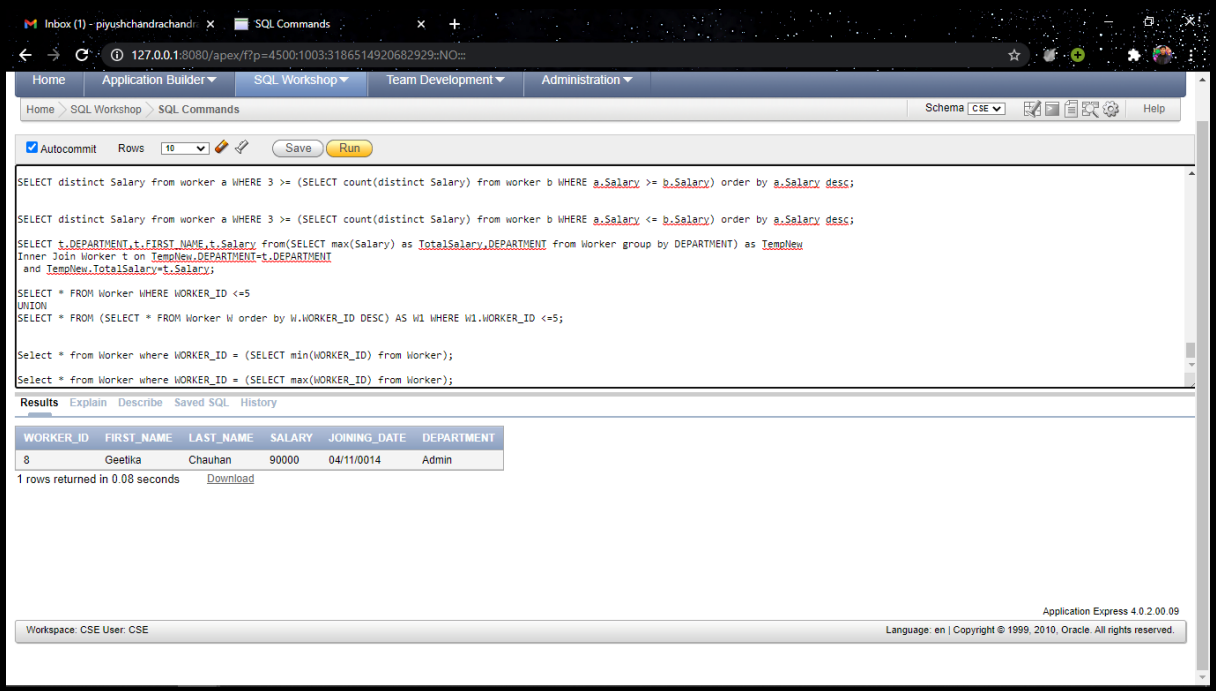
41. Write An SQL Query To Show All Departments Along With The Number Of People In There from Worker Table.

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



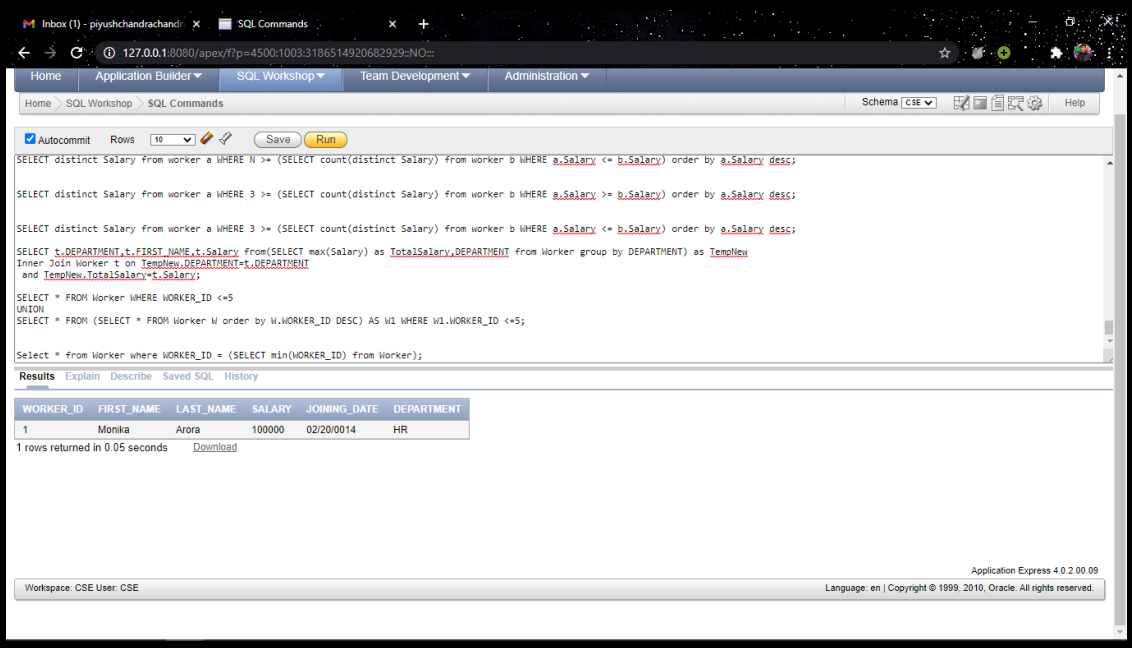
42. Write An SQL Query To Show The Last Record From Worker Table.

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



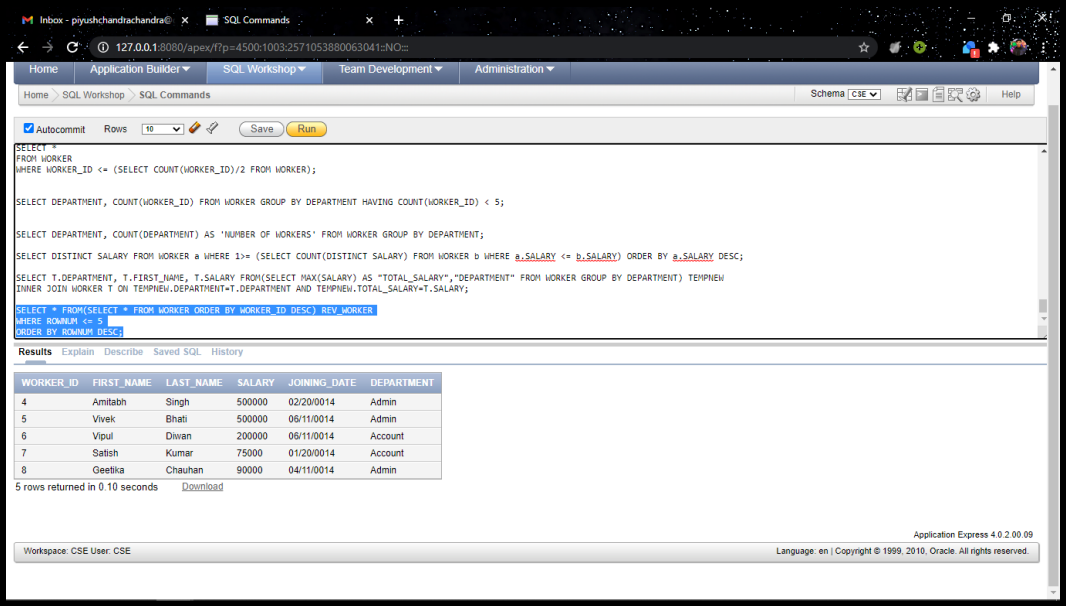
43. Write An SQL Query To Fetch The First Row Of Worker Table.

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



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

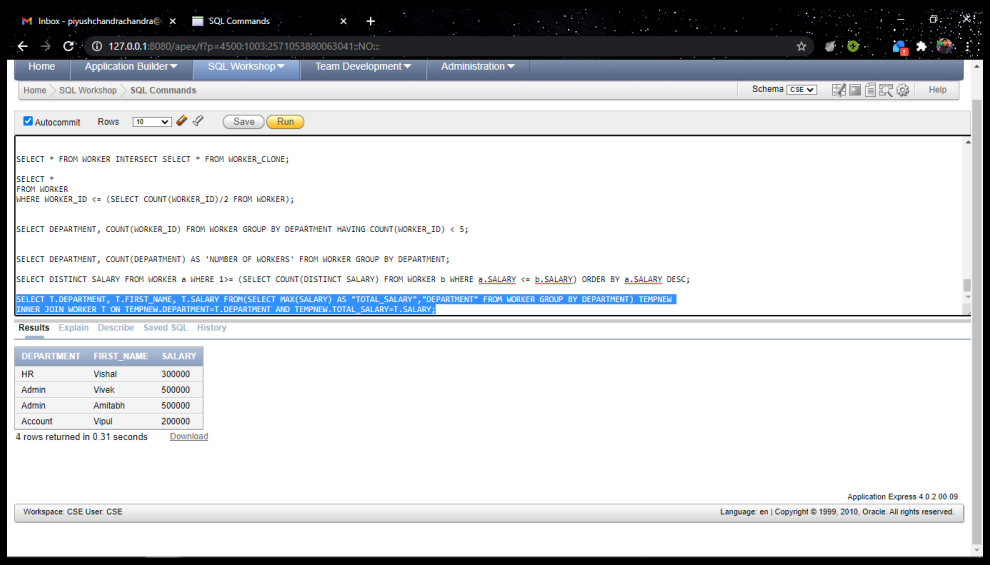
➢ SELECT * FROM(SELECT * FROM WORKER ORDER BY WORKER_ID DESC) REV_WORKER
WHERE ROWNUM <= 5
ORDER BY ROWNUM DESC;



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

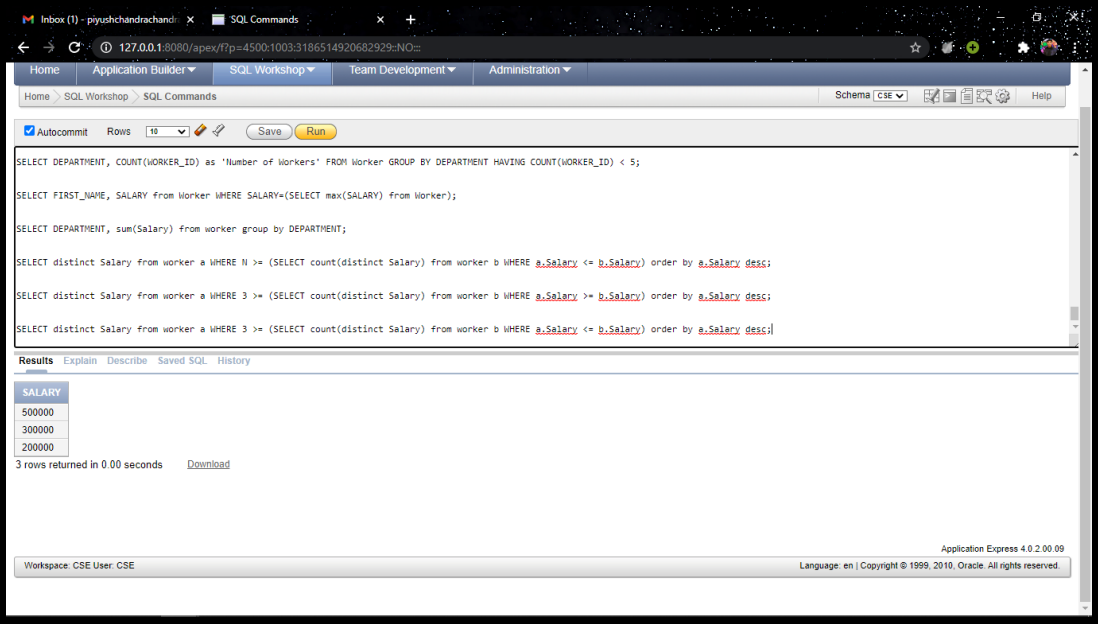
- SELECT T.DEPARTMENT, T.FIRST_NAME, T.SALARY FROM(SELECT MAX(SALARY) AS "TOTAL_SALARY","DEPARTMENT" FROM WORKER GROUP BY DEPARTMENT) TEMPNEW

  INNER JOIN WORKER T ON TEMPNEW.DEPARTMENT=T.DEPARTMENT AND TEMPNEW.TOTAL_SALARY=T.SALARY;



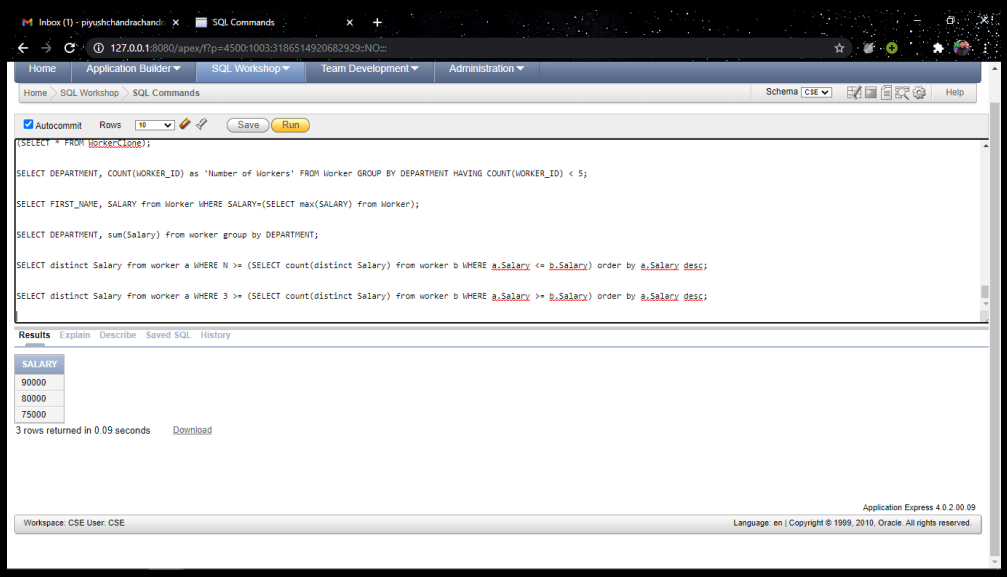
46. Write An SQL Query To Fetch Three Max Salaries From Worker Table.

- 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;



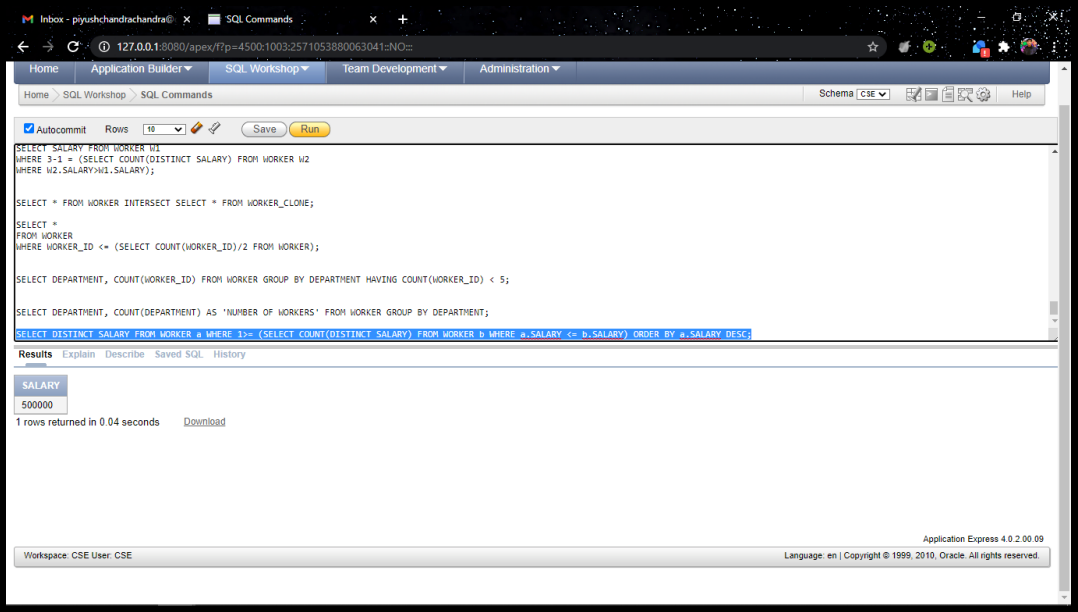
47. Write An SQL Query To Fetch Three Min Salaries From Worker Table.

- 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;



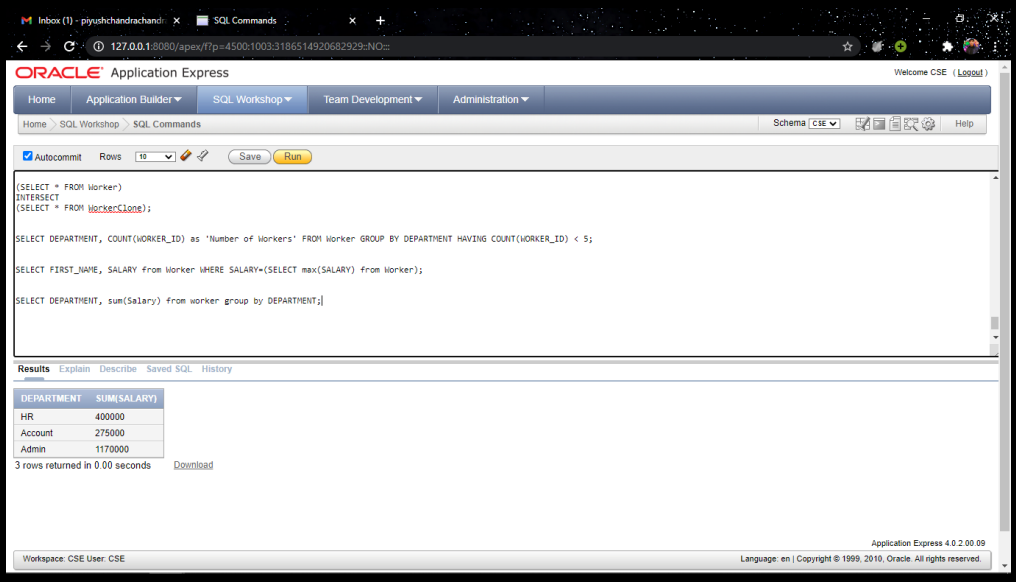
48. Write An SQL Query To Fetch Nth Max Salaries From Worker Table.

- 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;



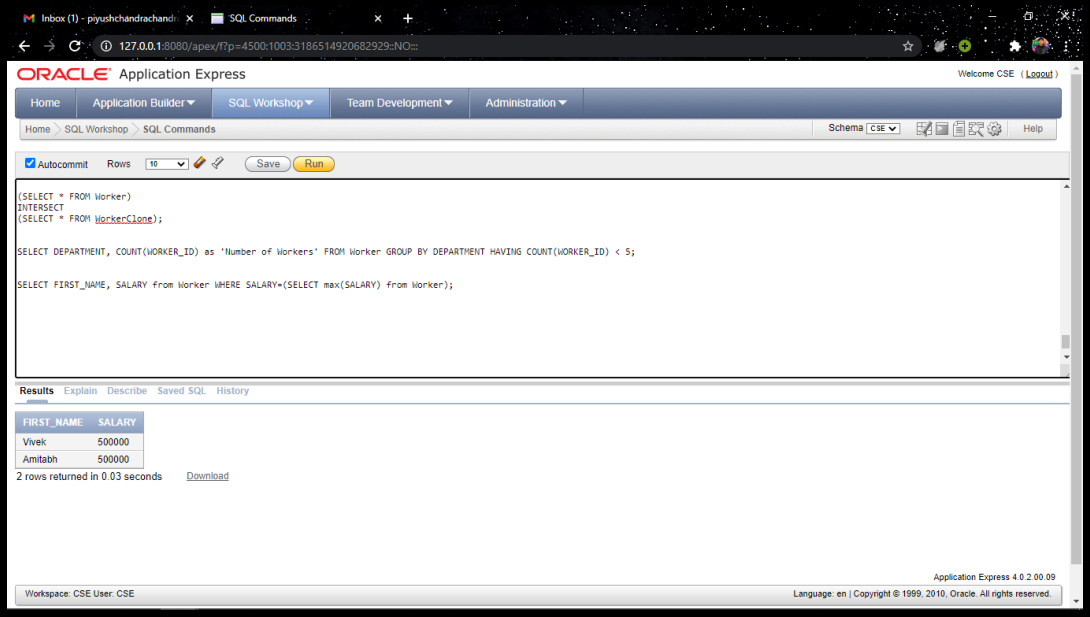
49. Write An SQL Query To Fetch Departments Along With The Total Salaries Paid For Each Of Them from Worker Table.

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



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);



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_rcv	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),

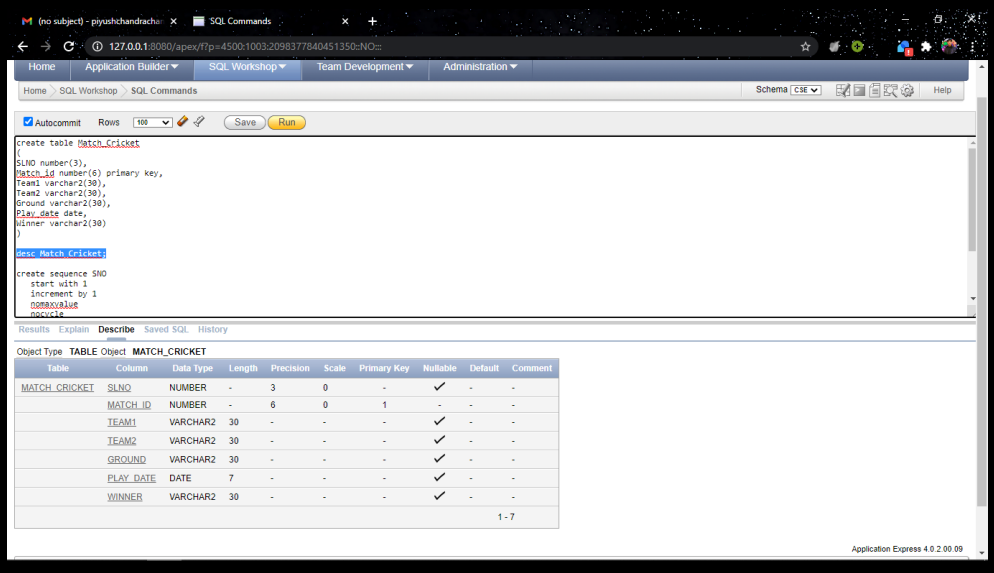
  Ground varchar2(30),

  Play_date date,

  Winner varchar2(30)

  );

  desc Match_Cricket;



- create table Player_Cricket(

  SLNO number(3),

  Player_id number(10) primary key,

  Lname varchar2(30),

  Fname varchar2(30),

  Country varchar2(30),

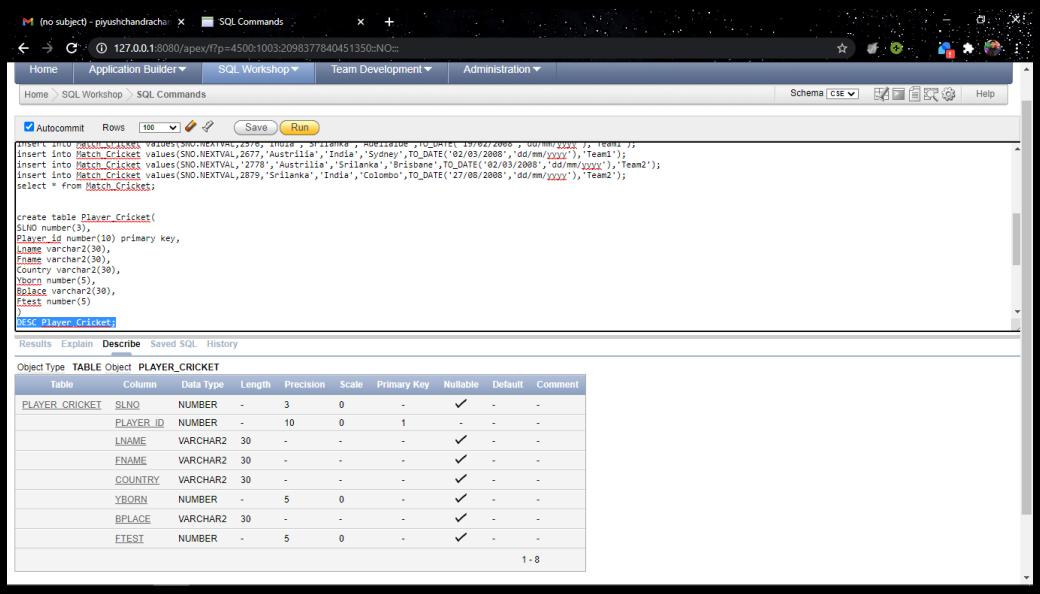
  Yborn number(5),

  Bplace varchar2(30),

  Ftest number(5)

)

  DESC Player_Cricket;



- 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),

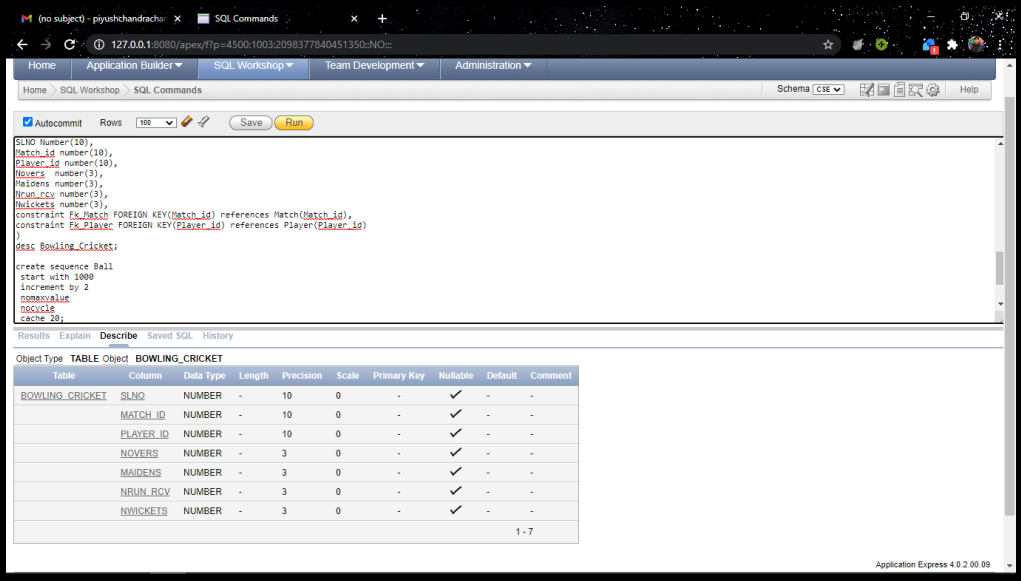
  Nwickets number(3),

  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;



- create table Batting_Cricket(

  SLNO number(7),

  Match_id number (10),

  Player_id number(10),

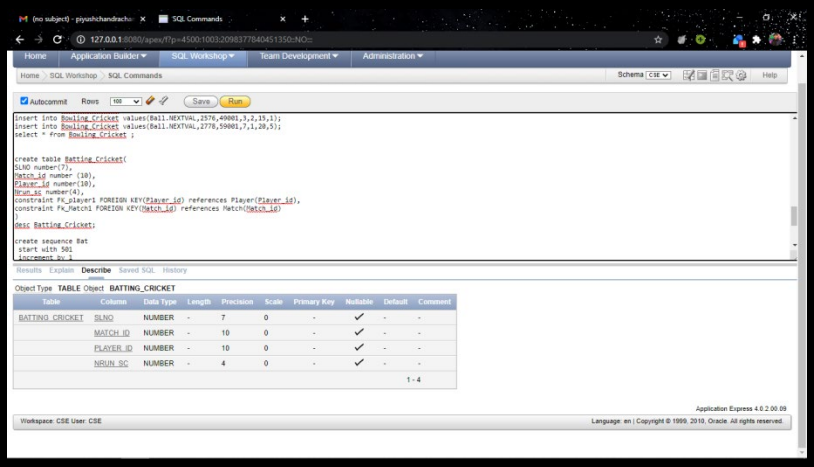
  Nrun_sc number(4),

  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;



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_rcv	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	Mattumagala	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');

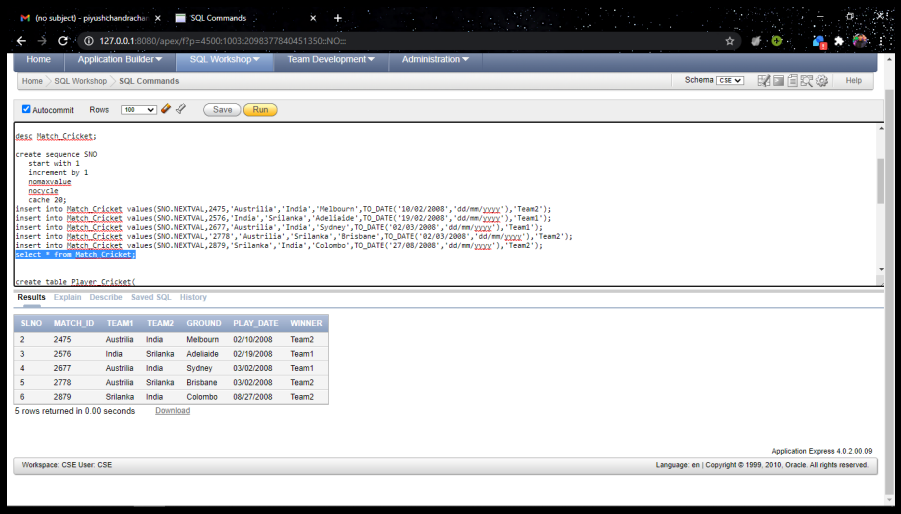
  insert into Match_Cricket values(SNO.NEXTVAL,2576,'India','Srilanka','Adeliaide',TO_DATE('19/02/2008','dd/mm/yyyy'),'Team1');

  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;



- insert into Player_Cricket
values(SP.NEXTVAL,69002,'Symonds','Andrew','Austrilia',1975,'Birminghum','2004');

insert into Player_Cricket
values(SP.NEXTVAL,69001,'Lee','Brett','Austrilia',1976,'Wollongong',1999);

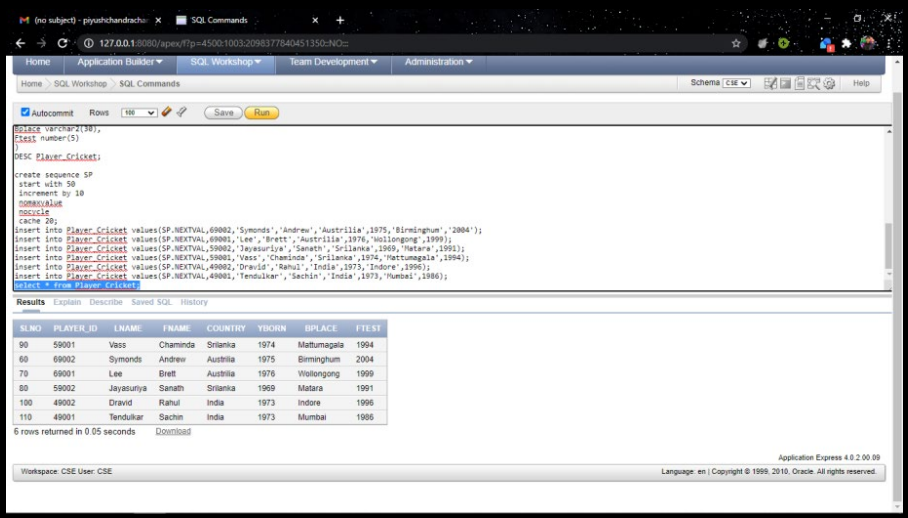
insert into Player_Cricket
values(SP.NEXTVAL,59002,'Jayasuriya','Sanath','Srilanka',1969,'Matara',1991);

insert into Player_Cricket
values(SP.NEXTVAL,59001,'Vass','Chaminda','Srilanka',1974,'Mattumagala',1994);

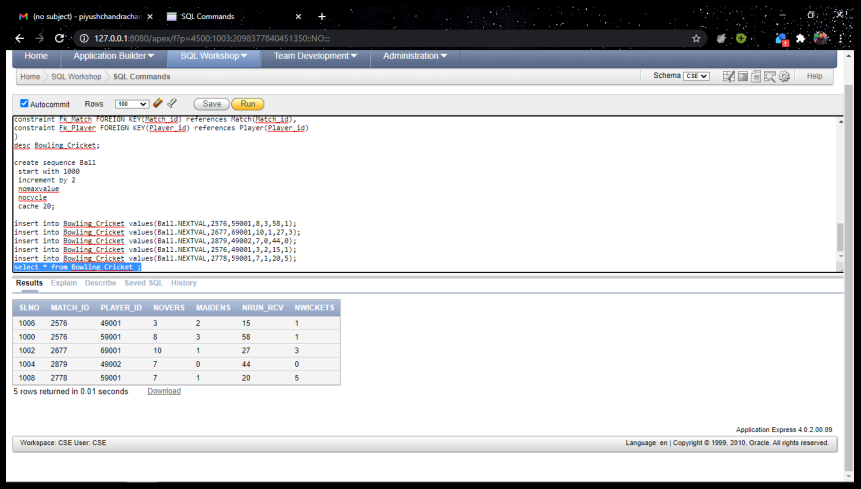
insert into Player_Cricket
values(SP.NEXTVAL,49002,'Dravid','Rahul','India',1973,'Indore',1996);

insert into Player_Cricket
values(SP.NEXTVAL,49001,'Tendulkar','Sachin','India',1973,'Mumbai',1986);

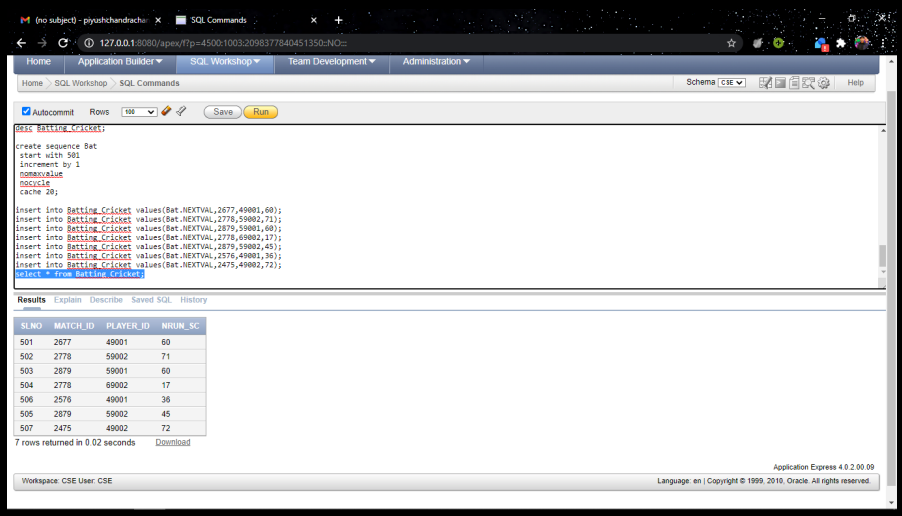
select * from Player_Cricket;



- 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;



- 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;



4.

a) Find the ground of the matches batted by player whose Fname is starting from 'S'.

b) Find Id's of player that have bowled in the match 2576 but not have batted.

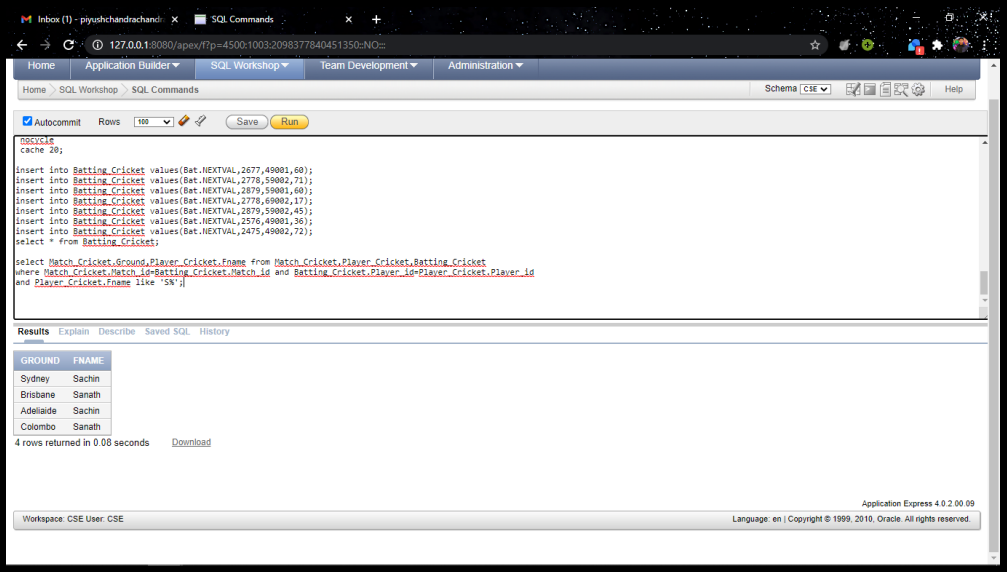
c) Find the batting average of each Indian player along with the Player_Id

d) 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%';



b) SELECT PLAYER_ID

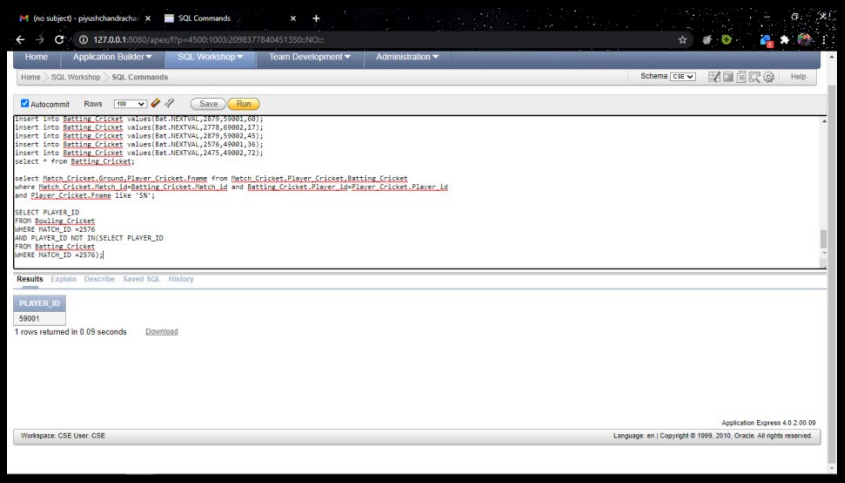
FROM Bowling_Cricket

WHERE MATCH_ID =2576

AND PLAYER_ID NOT IN(SELECT PLAYER_ID

FROM Batting_Cricket

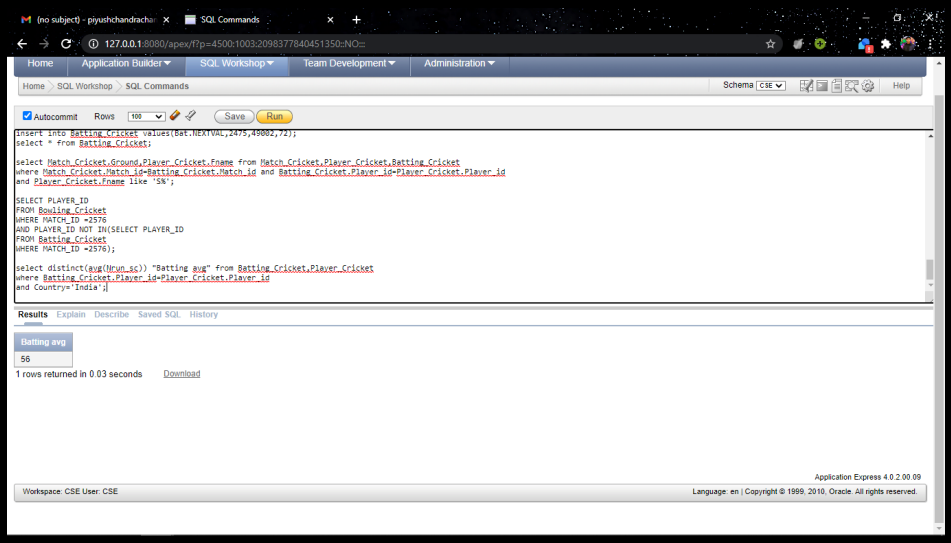
WHERE MATCH_ID =2576);



c) select distinct(avg(Nrun_sc)) "Batting avg" from Batting_Cricket,Player_Cricket

where Batting_Cricket.Player_id=Player_Cricket.Player_id

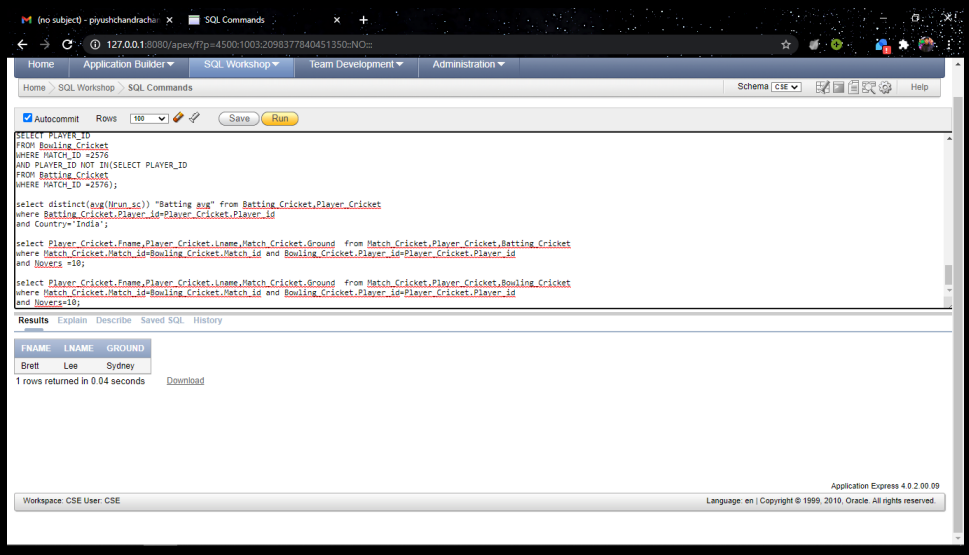
and Country='India';



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;



5.

a) Find the total run scored by a player who played First Test in 1991.

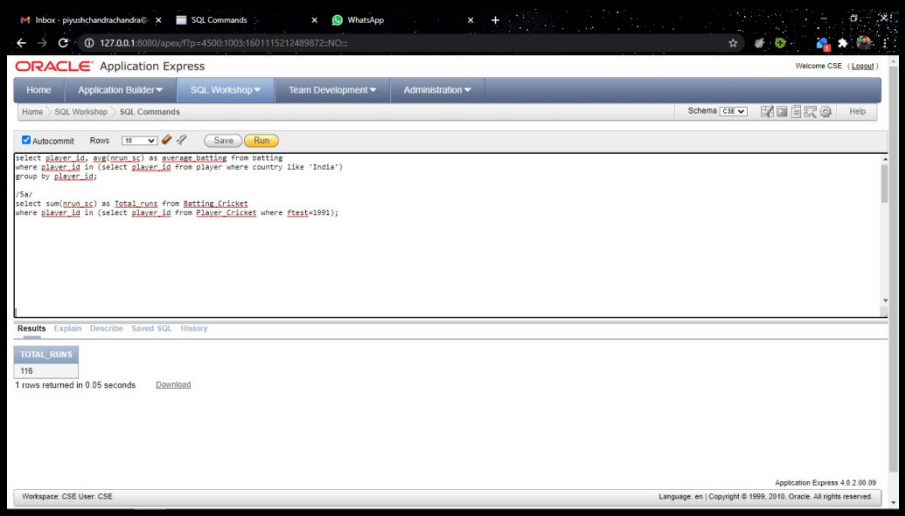
b) Find the name and the number of wickets taken by the youngest player in the data base.

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

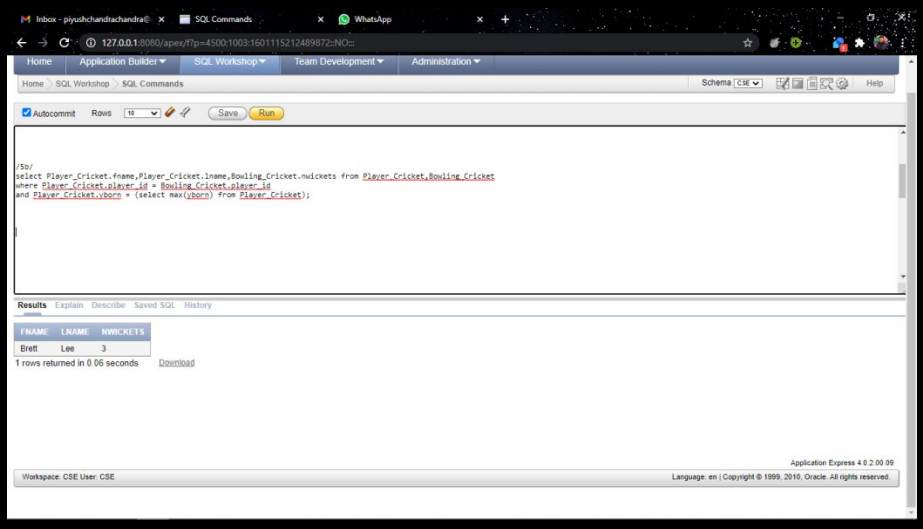
where player_id in (select player_id from Player_Cricket where ftest=1991);



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

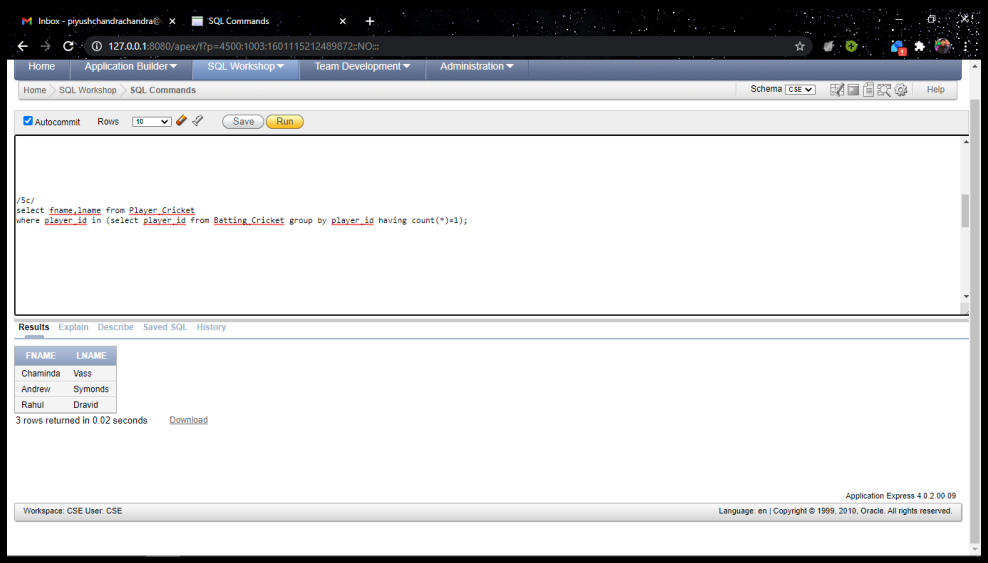
where Player_Cricket.player_id = Bowling_Cricket.player_id

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



c) select fname,lname from Player_Cricket

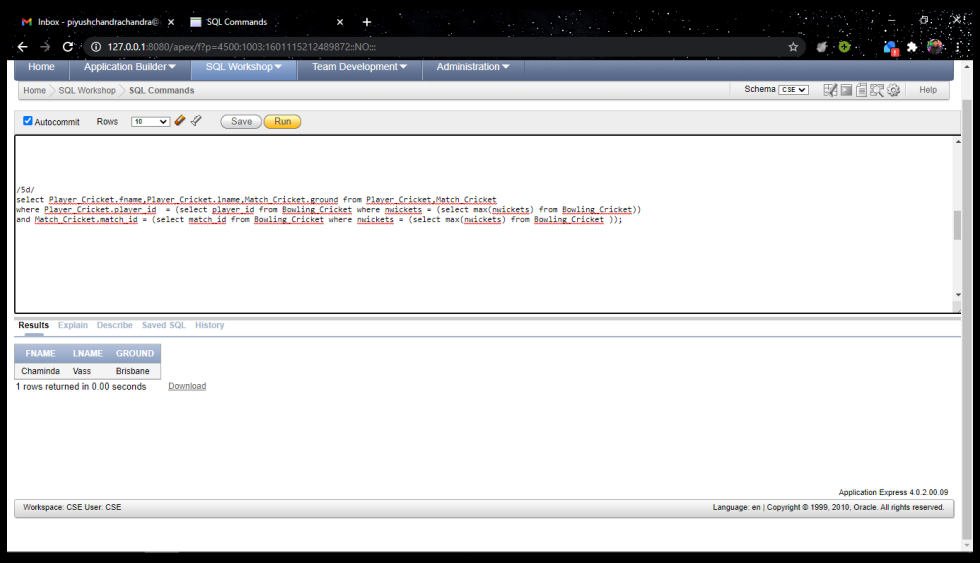
where player_id in (select player_id from Batting_Cricket group by player_id having count(*)=1);



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));



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

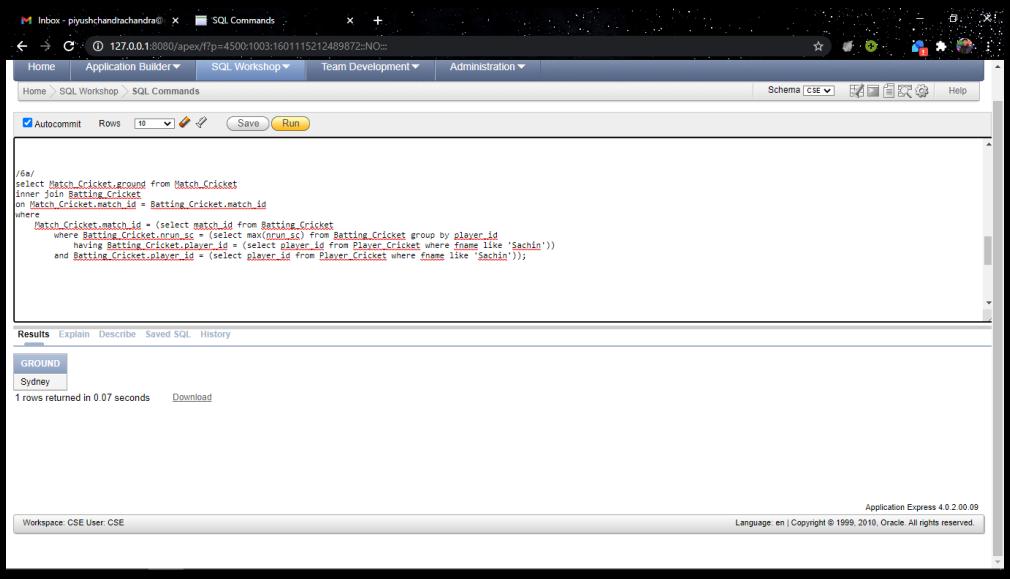
where

Match_Cricket.match_id = (select match_id from Batting_Cricket

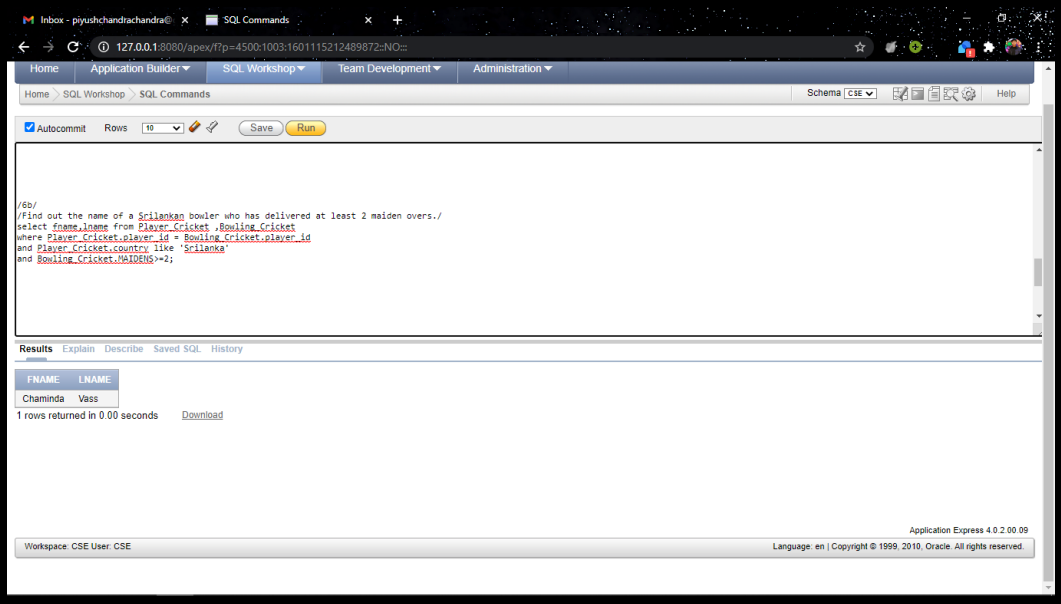
where Batting_Cricket.nrun_sc = (select max(nrun_sc) from Batting_Cricket group by player_id

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

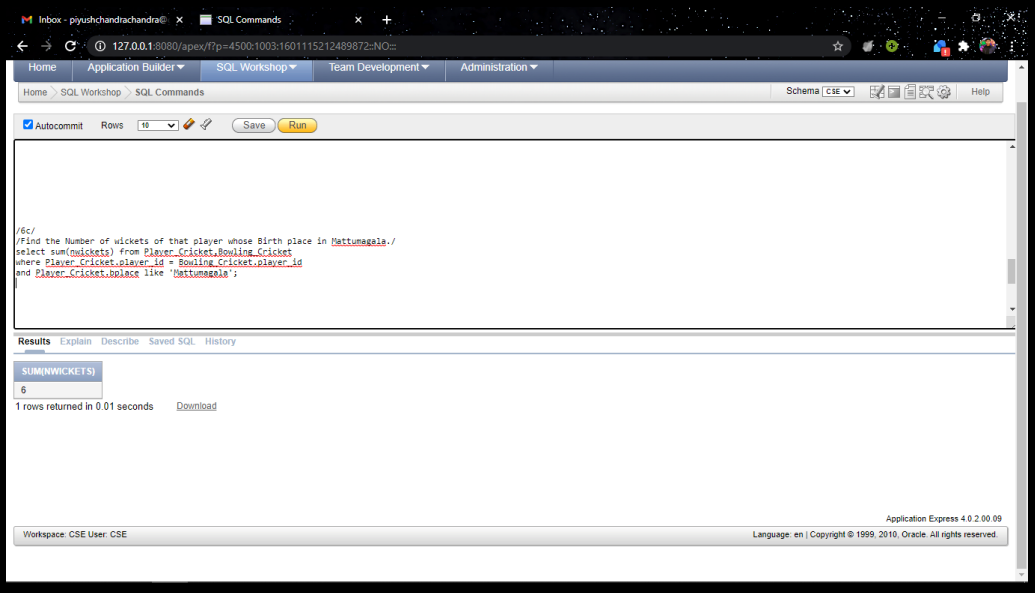
and Batting_Cricket.player_id = (select player_id from Player_Cricket where fname like 'Sachin'));



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;



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



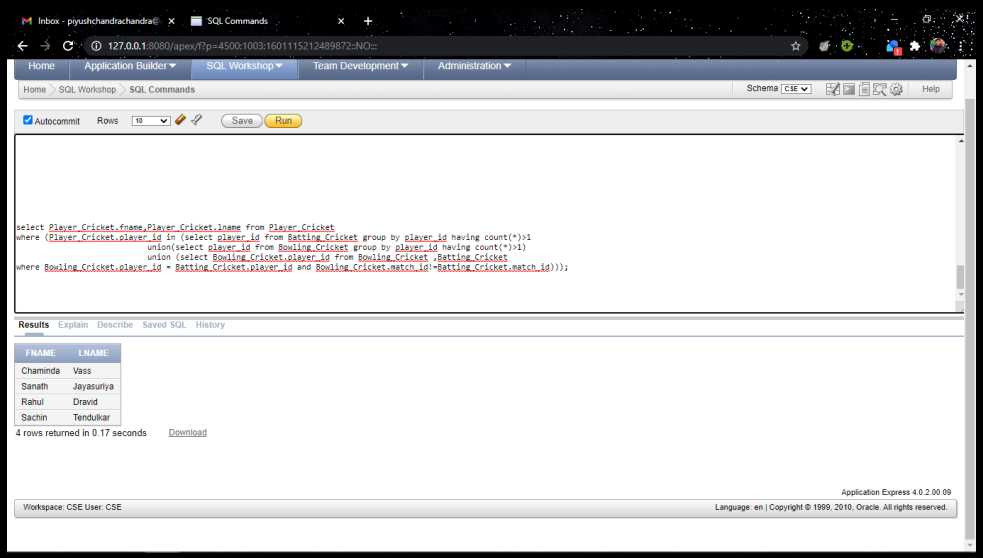
d) select Player_Cricket.fname,Player_Cricket.lname from Player_Cricket

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)));



THANK YOU

~~~~~~~~~~~~~~~~~~~~ X ~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~