## 2. Create the following tables and perform the queries that follow.

College

Column Name	Data type	Size
cName	varchar2	10
state	varchar2	10
enrollment	int	

Student

Column Name	Data type	Size
sID	int	
sName	varchar2	10
GPA	number	2,1
sizeHS	int	
DoB	date	

Apply

Column Name	Data type	Size
sID	int	
cName	varchar2	10
major	varchar2	20
decision	char	1

College Student

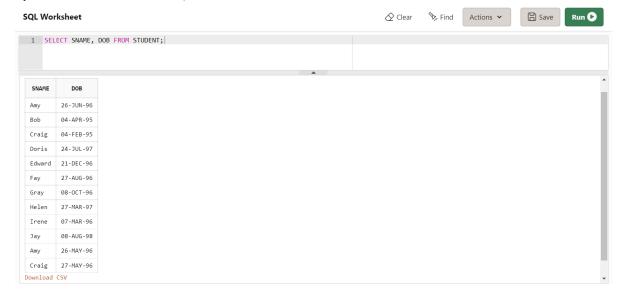
**Apply** 

CA	15000
CA	36000
MA	10000
NY	21000
MA	50040
CA	15000
CA	36000
MA	10000
NY	21000
MA	50040
GA	10012
	CA MA NY MA CA CA MA NY MA

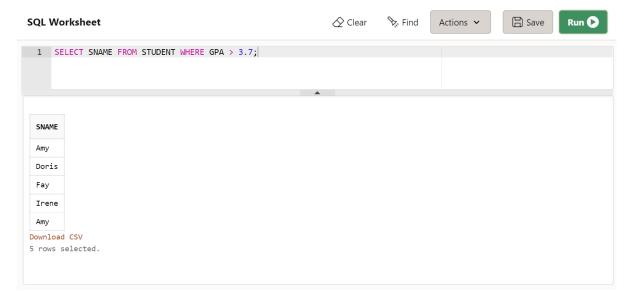
SID	SNAME	GPA	SIZEHS	DOB
122	Amy	3.9	1000	26-JUN-96
234	Bob	3.6	1500	04-APR-95
345	Craig	3.5	500	04-FEB-95
456	Doris	3.9	1000	24-JUL-97
567	Edward	2.9	2000	21-DEC-96
678	Fay	3.8	200	27-AUG-96
789	Gray	3.4	800	08-0CT-96
987	Helen	3.7	800	27-MAR-97
876	Irene	3.9	400	07-MAR-96
765	Јау	2.9	1500	08-AUG-98
654	Amy	3.9	1000	26-MAY-96
543	Craig	3.4	2000	27-MAY-96

SID	CNAME	MAJOR	DECISION
123	Stanford	cs	Υ
123	Stanford	EE	N
123	Berkeley	CS	Υ
123	Cornell	EE	Y
234	Berkeley	biology	N
345	MIT	bioengineering	Υ
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
678	Stanford	hitory	Y
987	Stanford	CS	Y
987	Berkeley	CS	Y
876	Stanford	CS	N
765	Stanford	history	Υ
765	Cornell	history	N
765	Cornell	psychology	Y
543	MIT	CS	N

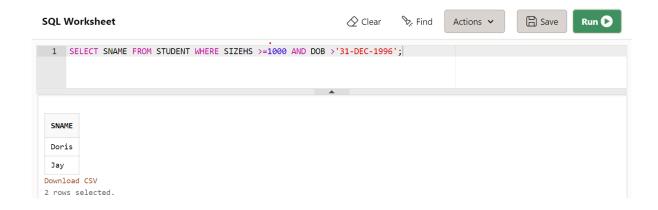
a) List the student name, dob from student table.



b) List the name of students scoring more than 3.7 in GPA.



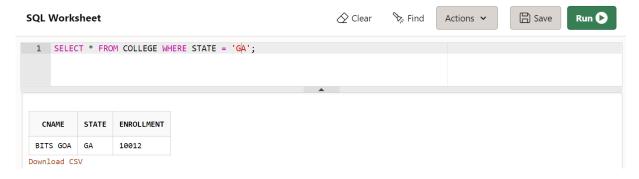
c) List the name of students whose High School size is at least 1000 and born after 1996. [Hint: check DoB greater than 31st December, 1996]



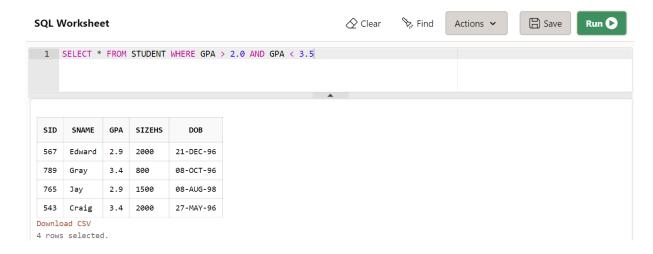
d) List the name of students who are scoring GPA in between 2.9 and 3.9.



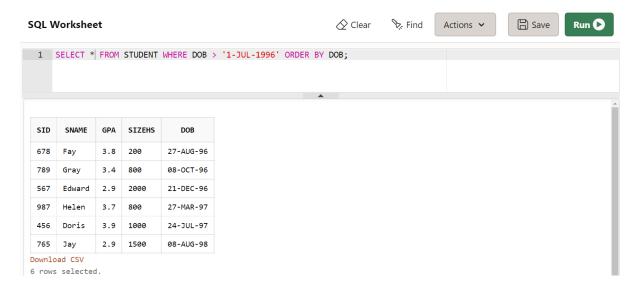
e) List all the details of colleges who situated in Goa.



f) List the students who have scored more than 2.0 but less than 3.5.



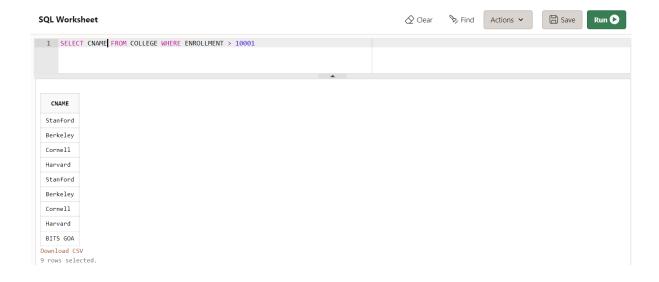
g) List the students who were born after 1st Jul 96 in the order of the Date of Birth.



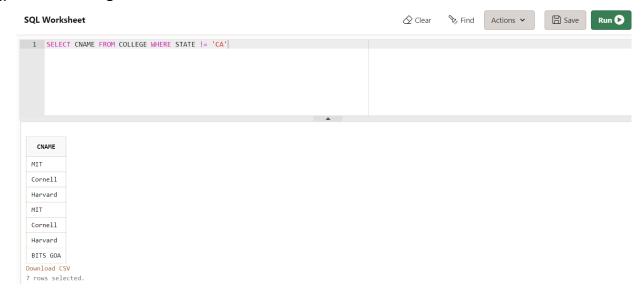
h) List the sID, cName, decision of applications that are accepted.



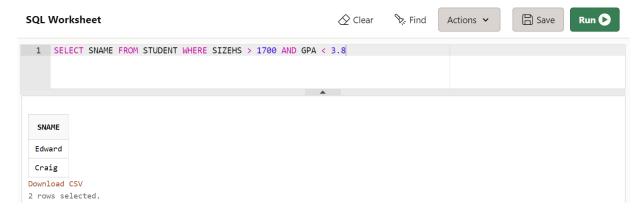
i) List the colleges that has enrolment greater than 10001.



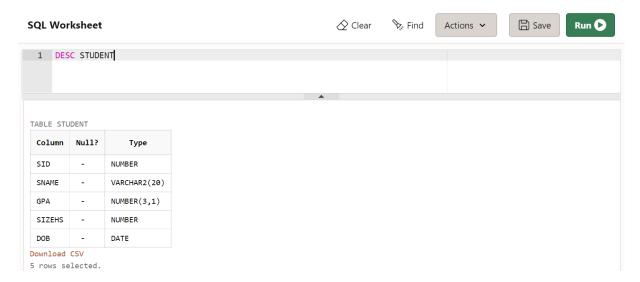
j) List the colleges not in California.



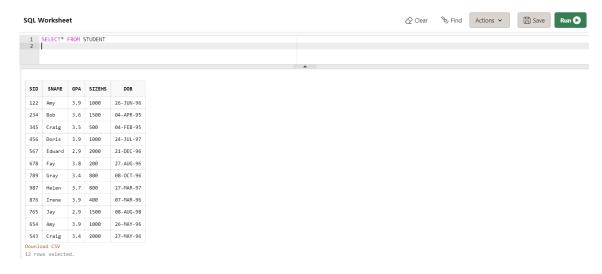
k) List names of all student who came from high school having size greater than 1700 and scored GPA less than 3.8.



I) Display the description of the Student table.



m) Display the details of all students.



n) Display unique majors.



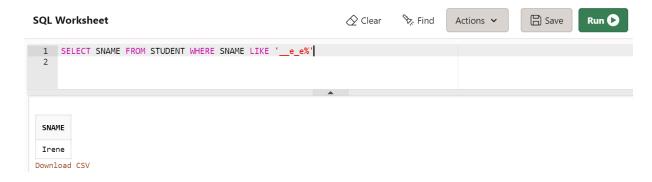
o) List the student names those are having five characters in their Names.



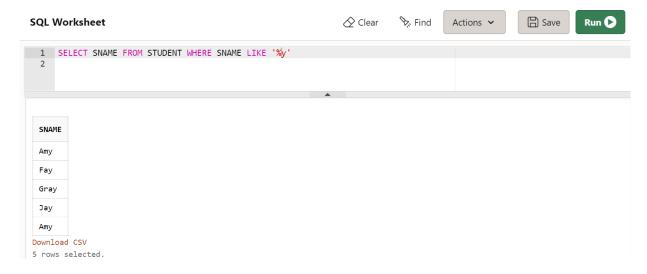
p) List the student names those are starting with 'H' and with five characters.



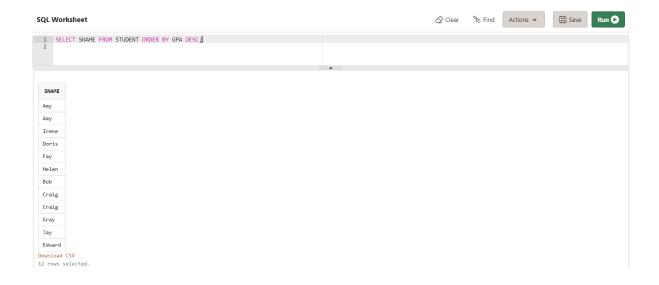
q) List the student names those are having third character and fifth char as 'e'.



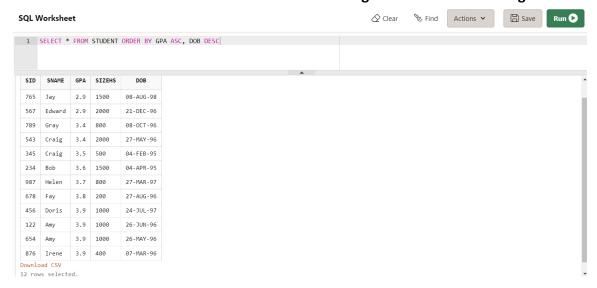
r) List the student names ending with 'y'



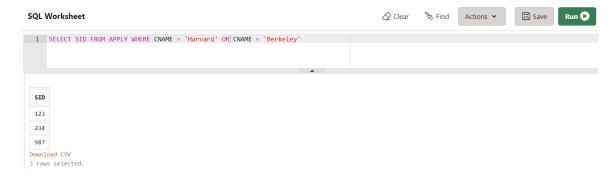
s) List the Students in the order of their GPA.



t) List the details of the students in order of the ascending of GPA and descending of DoB.



u) List the sIDs of student who apply in either 'Harvard' and 'Berkeley' college.



v) Delete all applications filled at Stanford (Choose table wisely)



w) Delete the college Stanford from college table.



x) Modify the GPA of all students by giving 10% raise in their GPA.



y) Increment the GPA of the students by 1.5 whose GPA is less than 3.5 and belongs to High School having size greater than 1500.

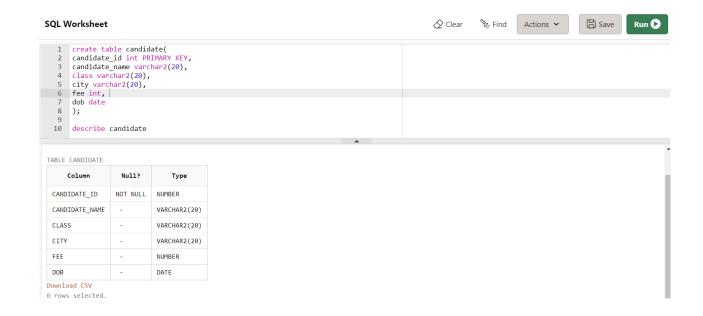


z) Delete the students who have scored less than 3.2 GPA.

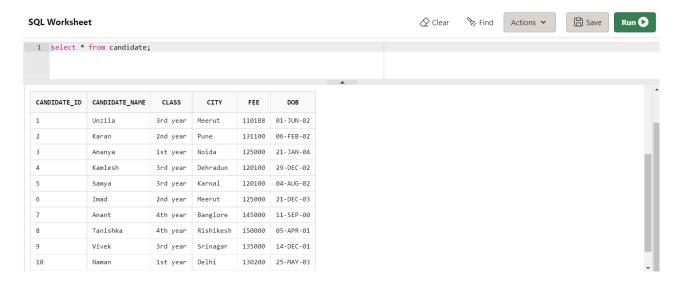


# 3. Execute the following queries: Consider the following schema: candidate (candidate\_id, candidate\_name, class, city, fee, DoB)

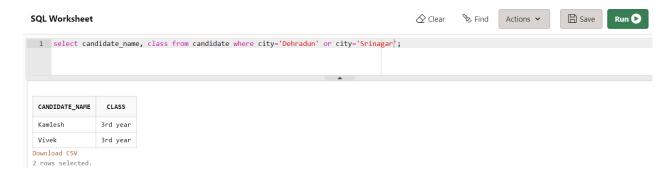
- Create a table named candidate where candidate\_id should be a primary key



- Insert 5 tuples into table.



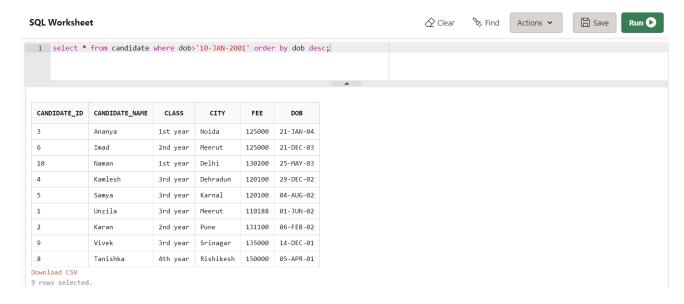
- Display the candidate names and class who either belong to Dehradun or Srinagar.



- Display the candidate\_id whose fee lies between 50000 - 85000.



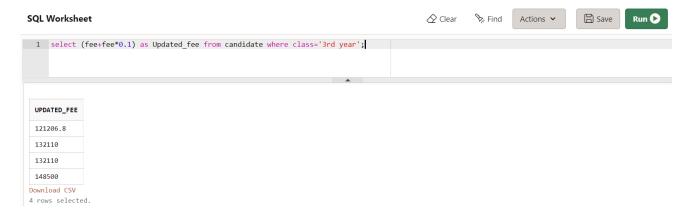
- Find the candidate details who were born after 10-Jan-2000 in descending order.



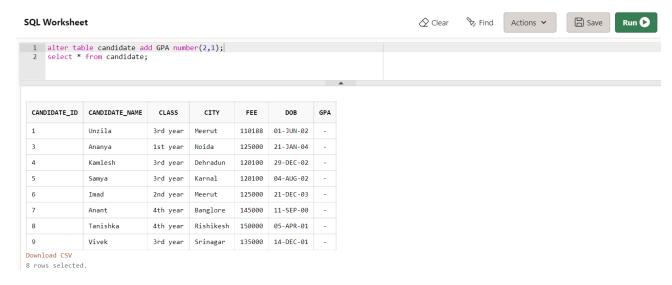
- Remove the details of the candidate whose name contains 'b' and ends with 'k'.



- Increase and display the fee all candidates of class '3rd year' by 10%.



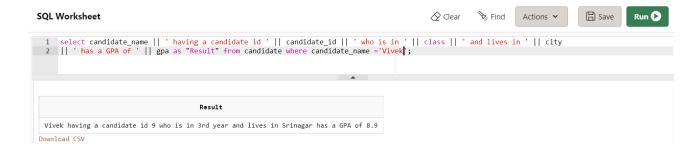
- Add a new column 'GPA' to the existing table.



- Display the following output:

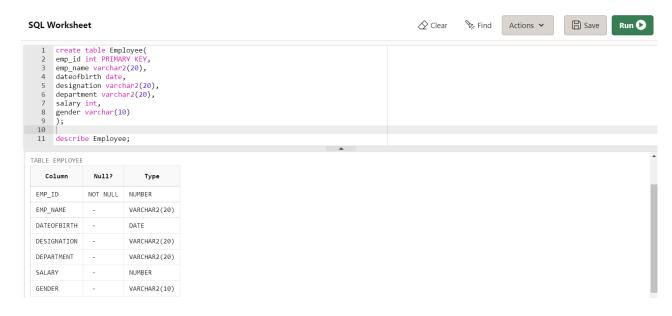
Sample output (in a single column, column name 'Result'):

Suraj having a candidate id 101 who studies in class IX and lives in Dehradun has a GPA of 7.

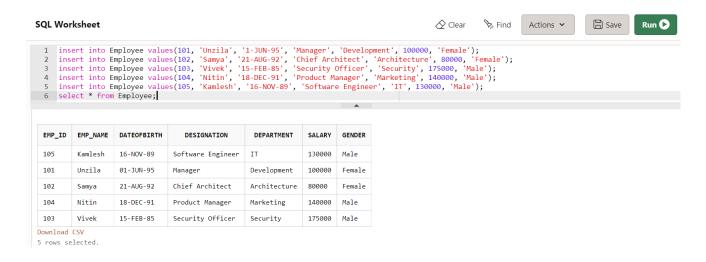


Database Schema for a Employee-Detail employee(emp\_id, emp\_name, dateofbirth, designation, department, salary, gender)

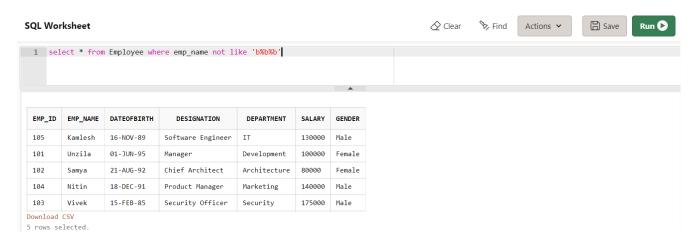
- Create the tables with the appropriate integrity constraints.



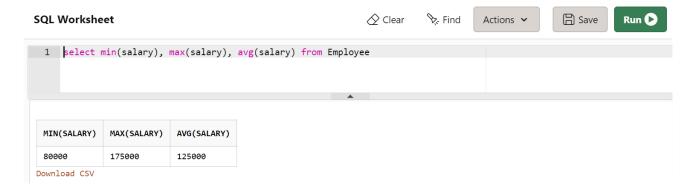
- Insert 5 records in each of the tables



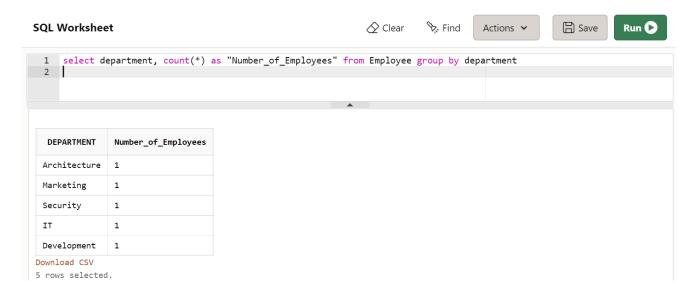
- List the employee details whose name does not contain 'b'.



- List the minimum, maximum, average salaries of employees.



- Display the number of employees working in each department and their department name (Using group by).



- List the Male programmers earning below the average salary of female programmers (Using group by).



-Who is the youngest programmer born in 1995(Using Nested query).



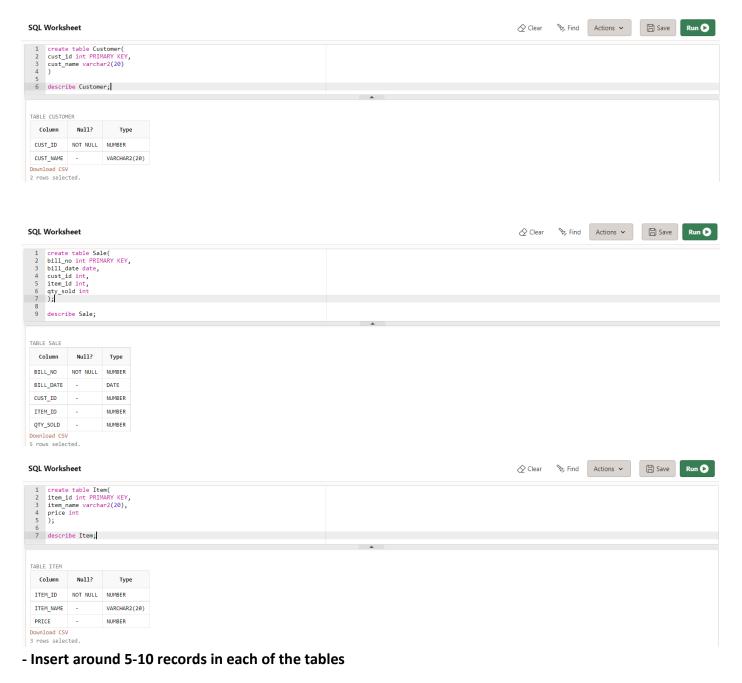
Database Schema for a customer-sale scenario

Customer(Cust id : integer, cust\_name: string)

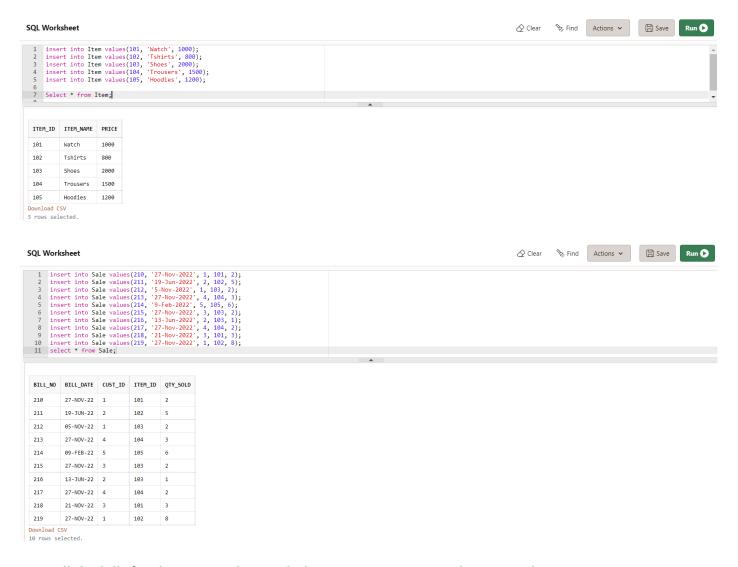
Item(item\_id: integer, item\_name: string, price: integer)

Sale(bill\_no: integer, bill\_date: date, cust\_id: integer, item\_id: integer, qty\_sold: integer)

- Create the tables with the appropriate integrity constraints



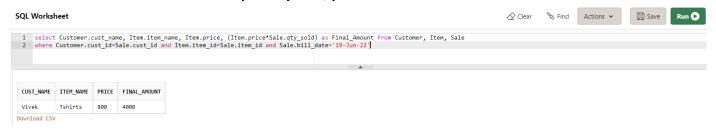




- List all the bills for the current date with the customer names and item numbers.



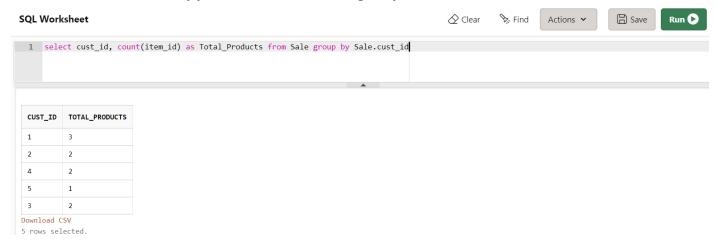
- List the total Bill details with the quantity sold, price of the item and the final amount



- List the details of the customer who have bought a product which has a price>1500



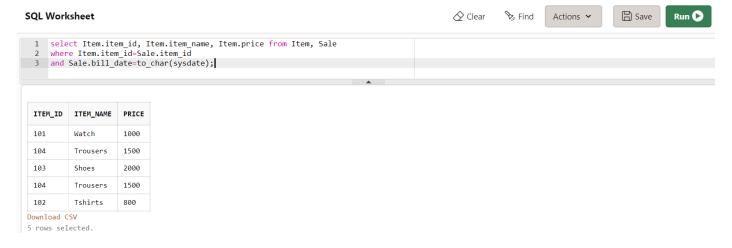
- Give a count of how many products have been bought by each customer.



- Give a list of products bought by a customer having cust\_id as 5



- List the item details which are sold as of today.

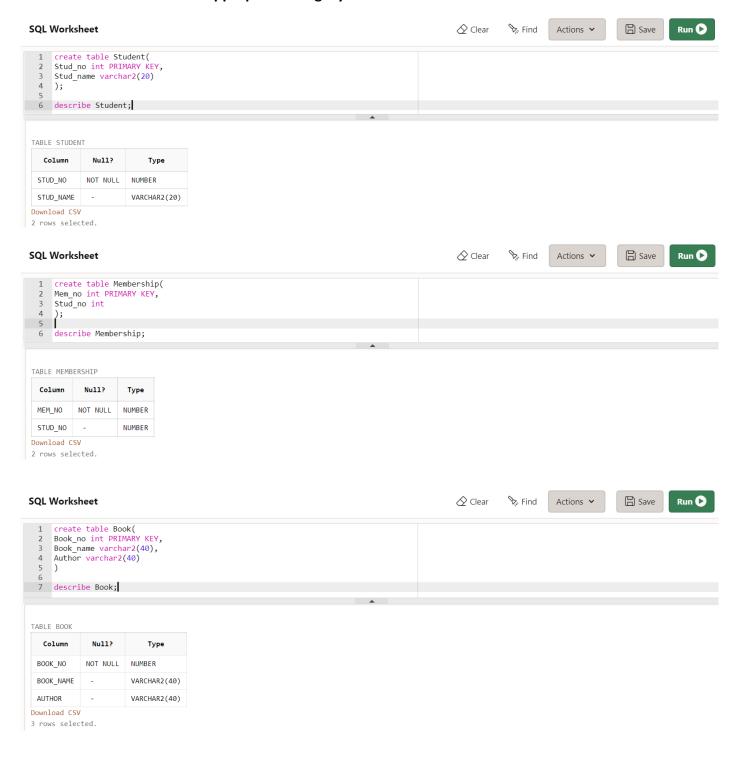


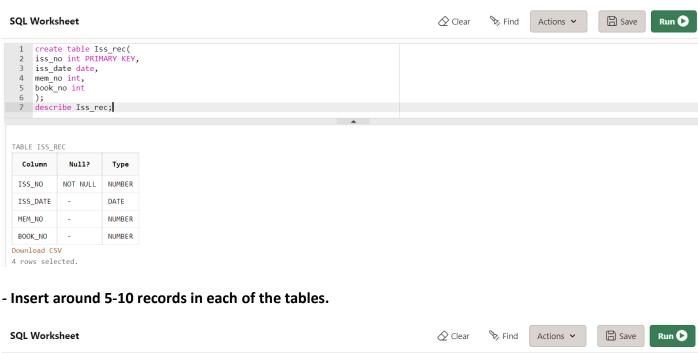
Database Schema for a Student Library scenario Student(Stud\_no: integer, Stud\_name: string) Membership(Mem\_no: integer, Stud\_no: integer)

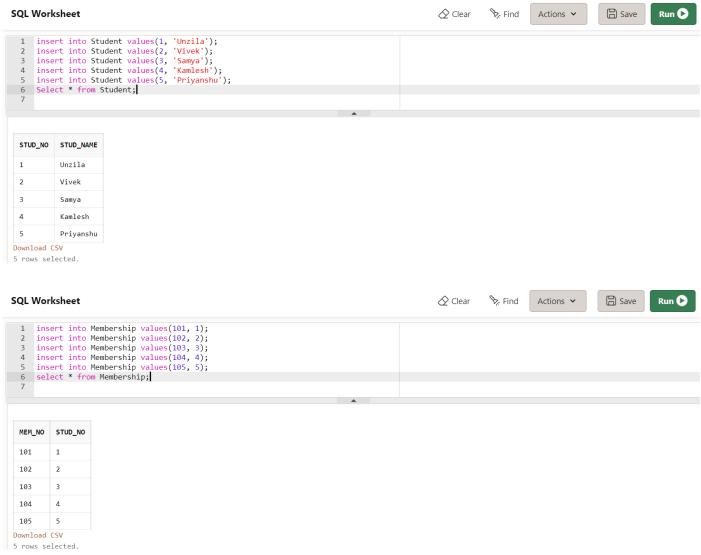
Book(book\_no: integer, book\_name:string, author: string)

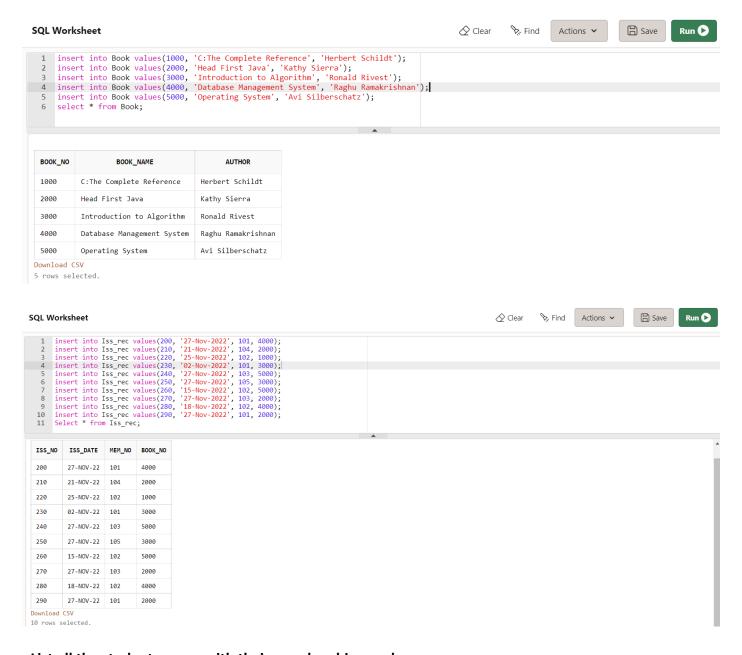
Iss\_rec(iss\_no:integer, iss\_date: date, Mem\_no: integer, book\_no: integer)

- Create the tables with the appropriate integrity constraints

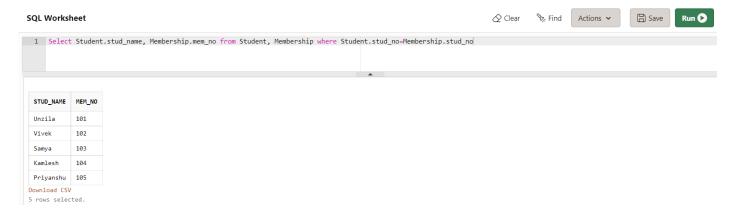




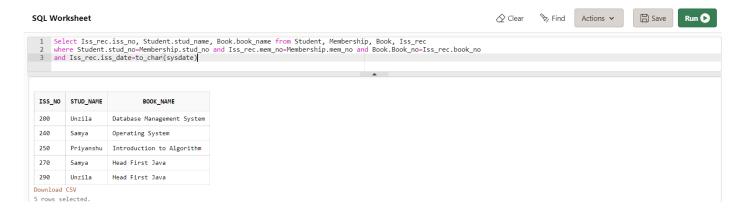




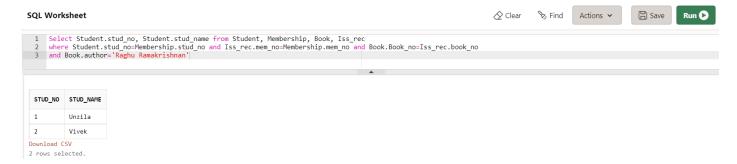
- List all the student names with their membership numbers.



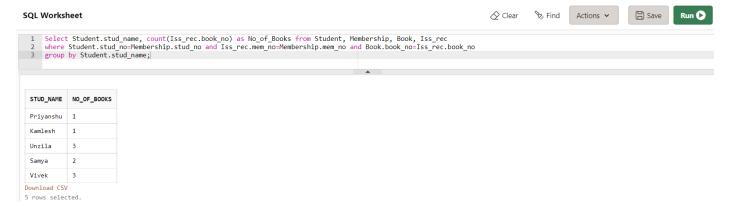
- List all the issues for the current date with student and Book names.



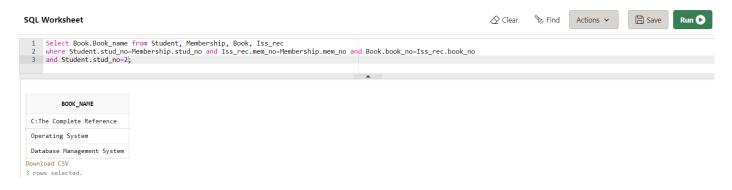
- List the details of students who borrowed book whose author is Raghu Ramakrishnan.



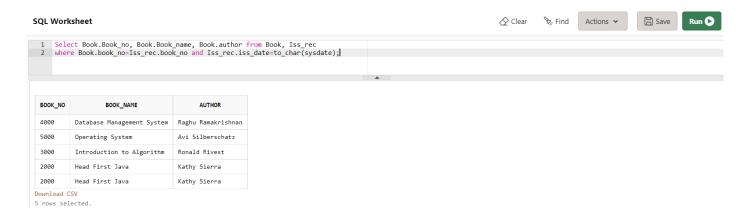
- Give a count of how many books have been bought by each student.



Give a list of books taken by student with stud\_no as 2



#### - List the book details which are issued as of today.



Database Schema for a Employee-pay scenario:

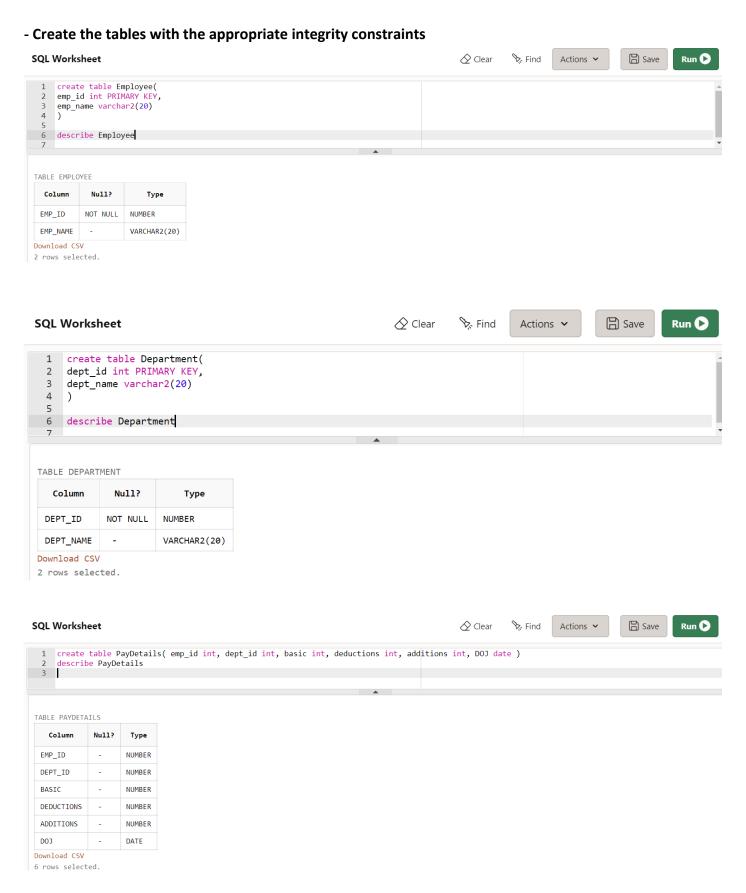
employee(emp\_id : integer, emp\_name: string)

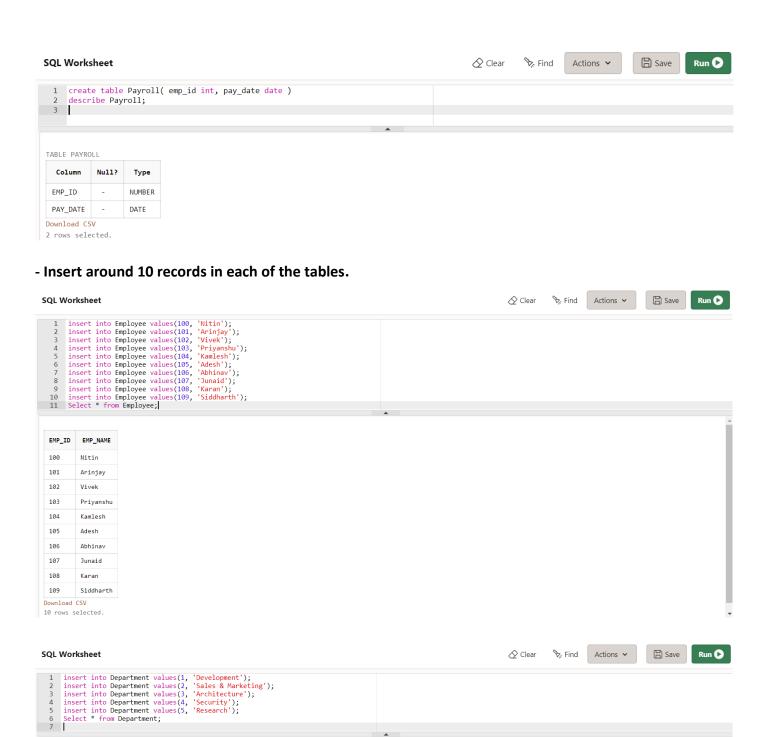
department(dept\_id: integer, dept\_name:string)

paydetails(emp\_id: integer, dept\_id: integer, basic: integer, deductions: integer, additions: integer, DOJ:

date)

payroll(emp\_id : integer, pay\_date: date)



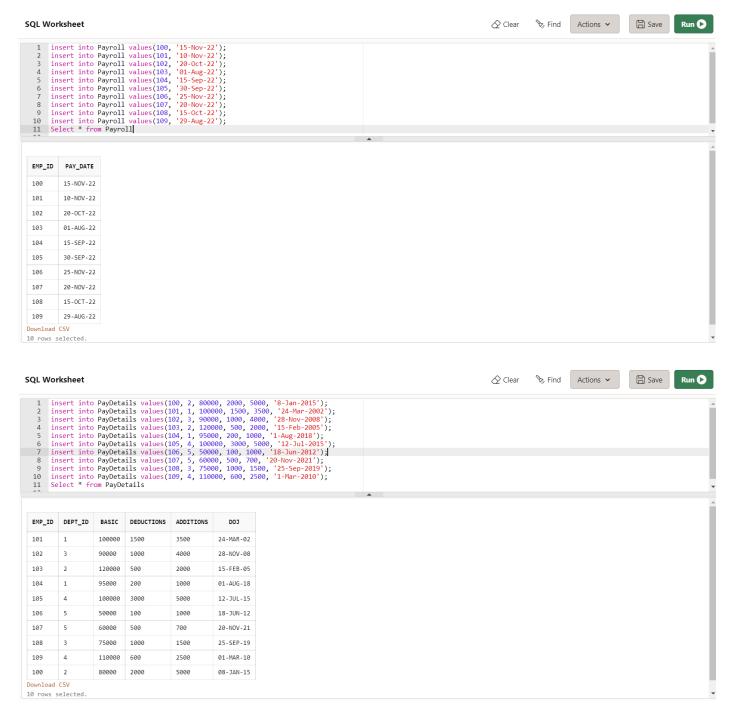


DEPT ID

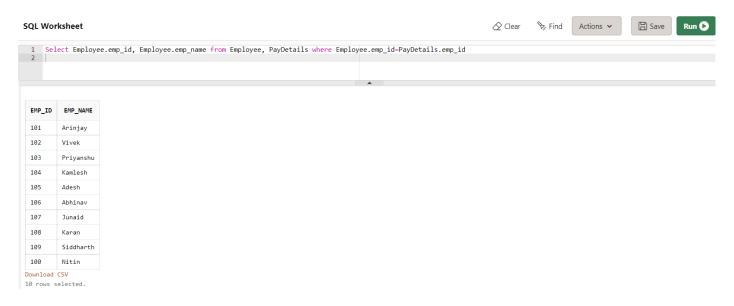
Download CSV 5 rows selected.

DEPT\_NAME
Development

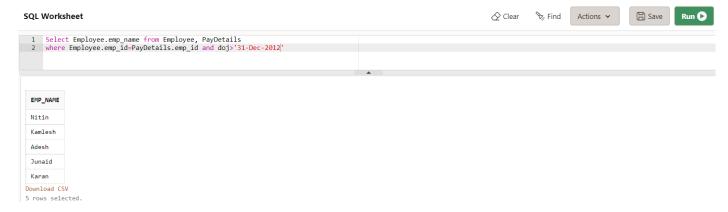
Architecture



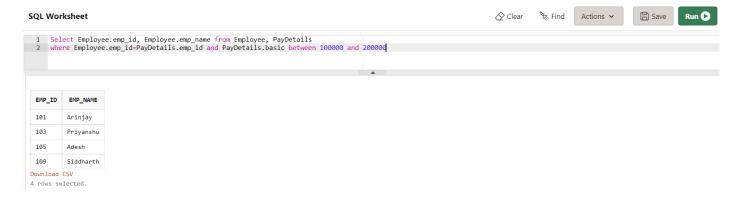
#### - List the employee details department wise.



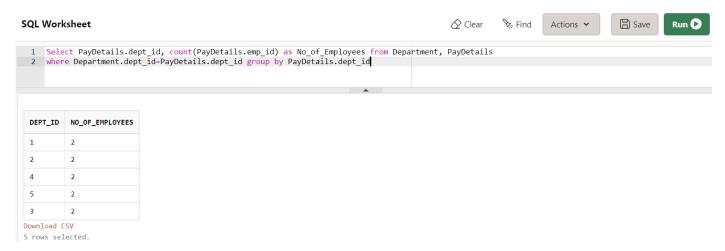
- List all the employee names who joined after particular date.



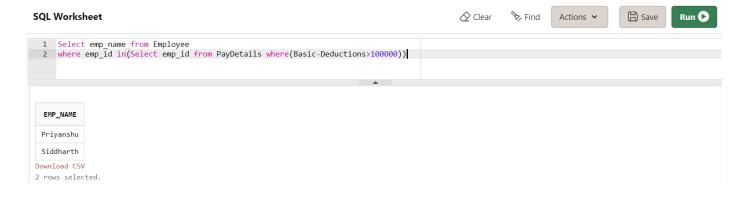
- List the details of employees whose basic salary is between 1,00,000 and 2,00,000.



- Give a count of how many employees are working in each department.



- Give a names of the employees whose netsalary>1,00,000.



### - List the details for an employee\_id=5



Database Schema for a Video Library scenario:

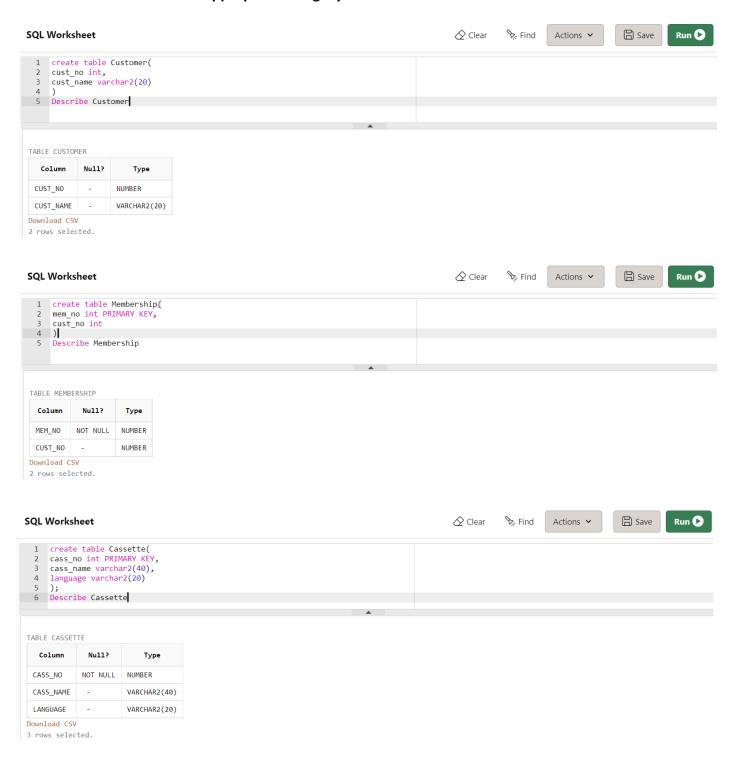
Customer(cust\_no: integer,cust\_name: string)

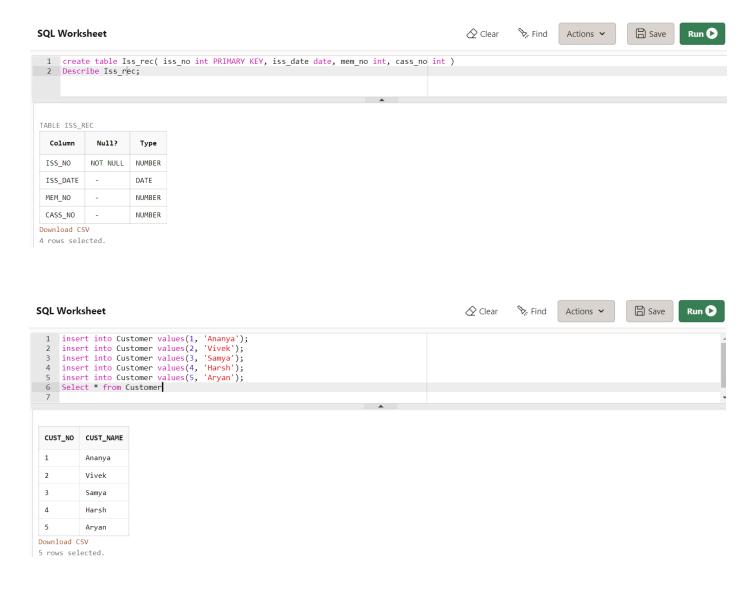
Membership(Mem\_no: integer, cust\_no: integer)

Cassette(cass\_no:integer, cass\_name:string, Language: String)

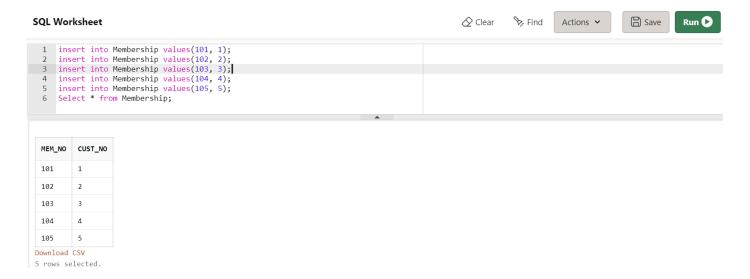
lss\_rec(iss\_no: integer, iss\_date: date, mem\_no: integer, cass\_no: integer)

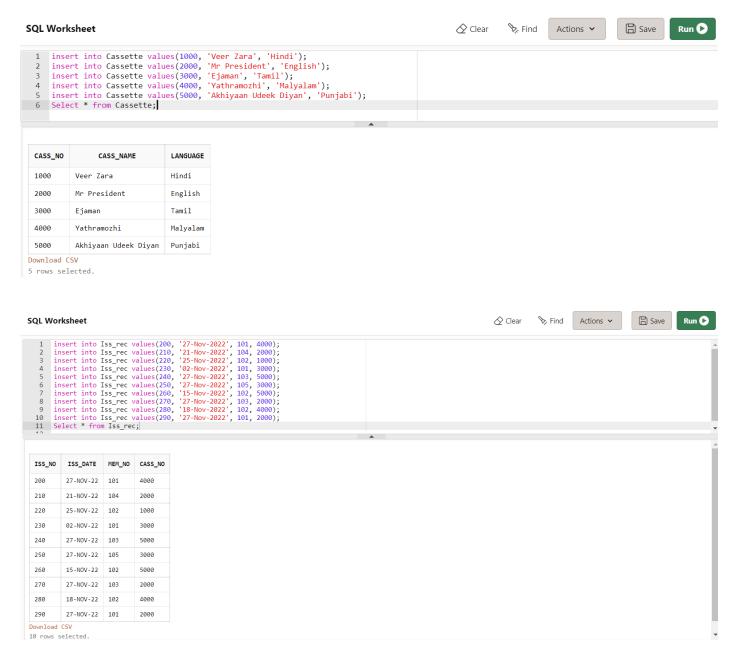
- Create the tables with the appropriate integrity constraints.



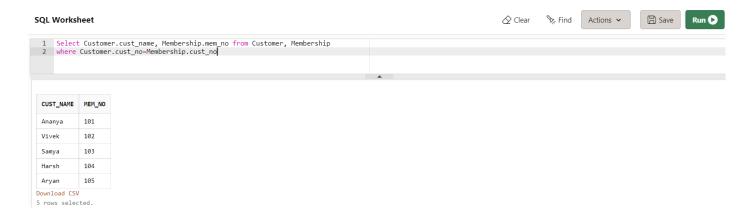


- Insert around 10 records in each of the tables.





- List all the customer names with their membership numbers.



- List all the issues for the current date with the customer names and cassette names.



- List the details of the customer who has borrowed the cassette whose title is "Veer Zara".



- Give a count of how many cassettes have been borrowed by each customer.



- Give a list of book which has been taken by the student with mem\_no as 5.



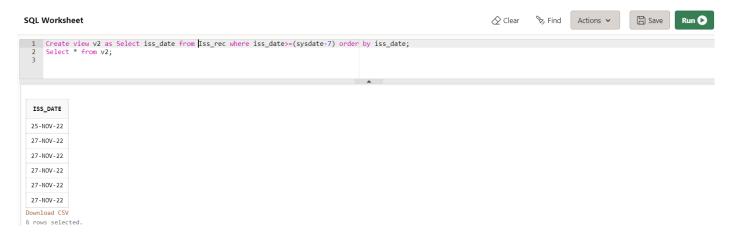
- List the cassettes issues for today



- Create a view which lists outs the iss\_no, iss\_date, cust\_name, cass\_name.



- Create a view which lists issues-date wise for the last one week.



Database Schema for a student-Lab scenario:

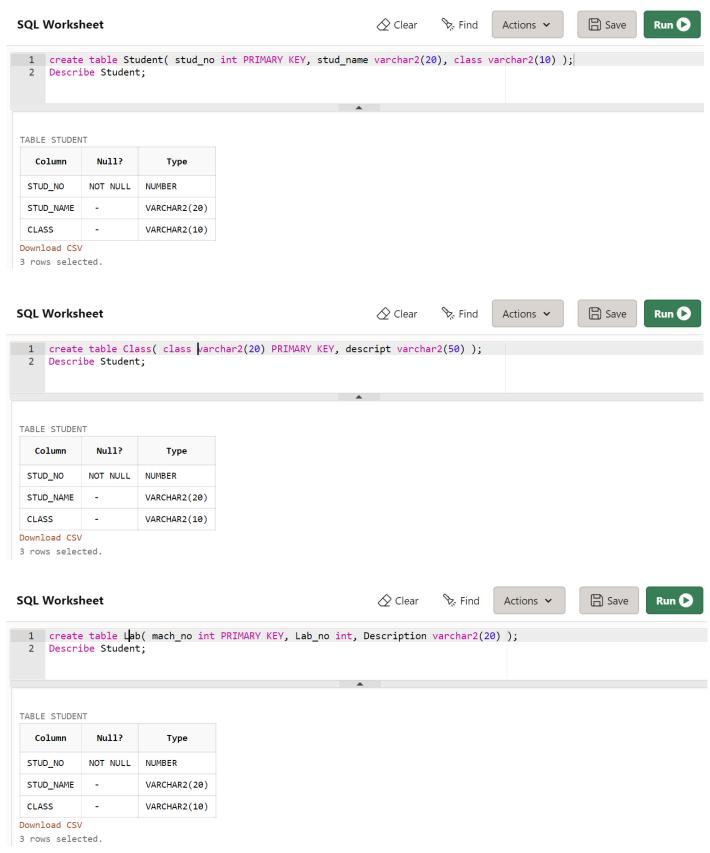
Student(stud\_no: integer, stud\_name: string, class: string)

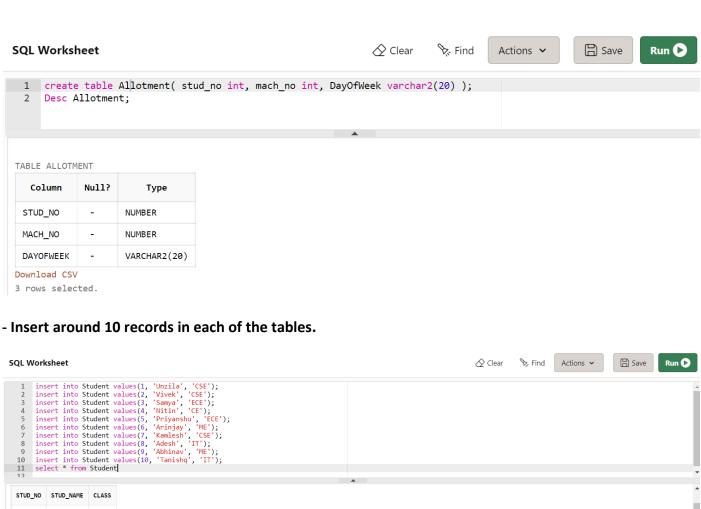
Class(class: string, descrip: string)

Lab(mach\_no: integer, Lab\_no: integer, description: String)

Allotment(Stud\_no: Integer, mach\_no: integer, dayof week: string)

- Create the tables with the appropriate integrity constraints.

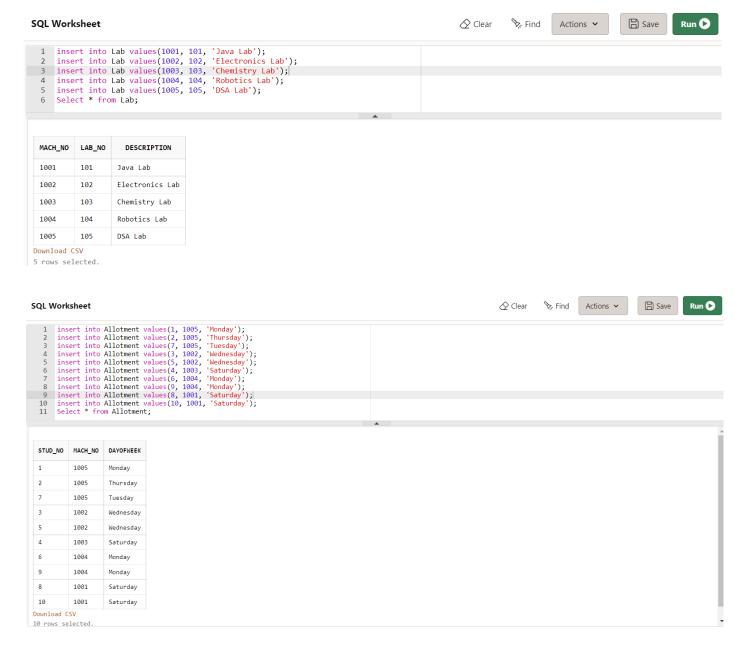




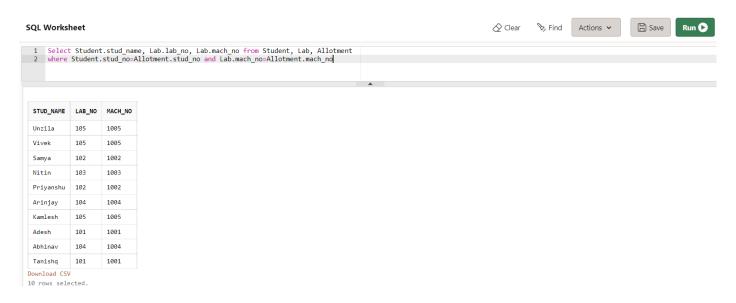




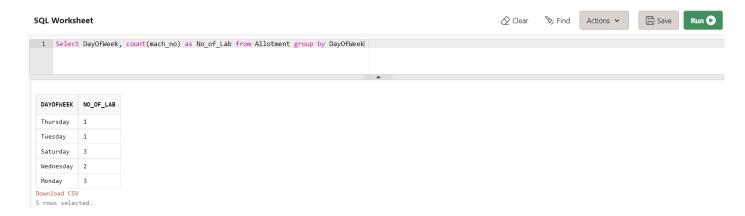
5 rows selected.



- List all the machine allotments with the student names, lab and machine numbers.



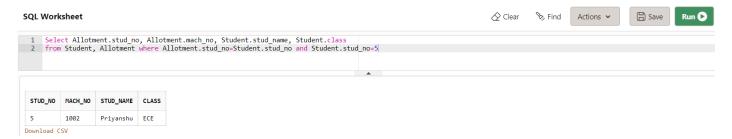
- List the total number of lab allotments day wise.



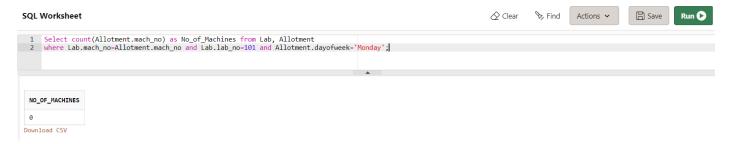
- Give a count of how many machines have been allocated to the 'CSE' class.



- Give a machine allotment details of the stud\_no 5 with his personal and class details.



- Count for how many machines have been allocated in Lab\_no 1 for the day of the week as "Monday".



- How many students class wise have allocated machines in the labs.

