S. D. M. COLLEGE OF EMPGINEERING AND TECHNOLOGY, DHARWAD – 580002

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING



**(An Autonomous Institution affiliated to Visvesvarayya Technological University, Belgavi )**

**Department of Information Science and Engineering**

As part of DBMS LAB PROJECT REPORT ON

**TRAVEL AGENCY MANAGEMENT SYSTEM**

Prepared By:

Chandana Hegde 2SD18IS014

Deepti Nayak 2SD18IS016

(Students of 5th sem,ISE 2020-21 Batch)

Under Guidance Of:

Prof. Leena I. Sakri

|  |
| --- |
| Contents of Report:  Table of contents  1. Project profile  2. Requirements  3. Schema diagram  4. ER Diagram  5. Normalizations forms 1, 2, 3 |

1. Project profile

|  |  |
| --- | --- |
| *Project definition* | *Travel agency Management System* |
| Objective | Main objective of this system is to provide online registration, Tour packages, online booking, online payment and facilities from quires. |
| Tools | M.S. World, ERDplus. |
| Internal Guide | Mrs. Leena Sakri |
| Submitted to | Department of information science |
| Developed by | Chandana Hegde  Deepti Nayak |

**TRAVEL AGENCY DATABASE**

Travel agency database shares details about the packages, customer, reservation, payment details and customer's reviews on the travel agency.

**Requirements:**

Travel agency offers one or more packages.

A customer can make zero or more reservation.

A customer selects one or more packages.

Customer makes payment according to reservation.

Customer may or may not review the agency.

Customer may have companions.

Reservation reveal rid, pname, transportation model, destination location, staying hotel, number of days, total price.

A customer has name, one or more given address, phone number, email, reservation\_no, amount\_paid, and a unique customer\_id, payment\_no, mode of Payment, amount and date of Payment.

Travel agency has a name, email and address.

**--Entities and attributes—**

1. CUSTOMER

Has following attributes.

* Name
* cid
* address
* phone
* email
* nationality
* gender
* age
* no\_of\_companions
* pno
* transaction \_mode
* amount\_paid
* reservation\_no
* date
* travel\_Agency\_name

2. TRAVEL AGENCY **:-**

Has following attributes.

* tname
* address
* email

3. RESERVATION:-

Has following attributes.

* rid
* Amount
* Start\_date
* End\_date
* Package\_name
* Location
* Hotel
* Package\_price
* Transportation
* No\_of\_days

Relationship between entities:

1. MAKES :

* A 1:1 relationship between CUSTOMER and RESERVATION.
* Participation of CUSTOMER is partial.
* Participation of RESERVATION is total.

2.REVIEW :

* A N:1 relationship between CUSTOMER and TRAVEL\_AGENCY
* Both of them have total participation.

**SCHEMA DIAGRAM**

**CUSTOMER**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | cid | pno | phone | address | nationality | gender | age | TransactionMode | tname | noofcompanion | Date | amount\_paid |

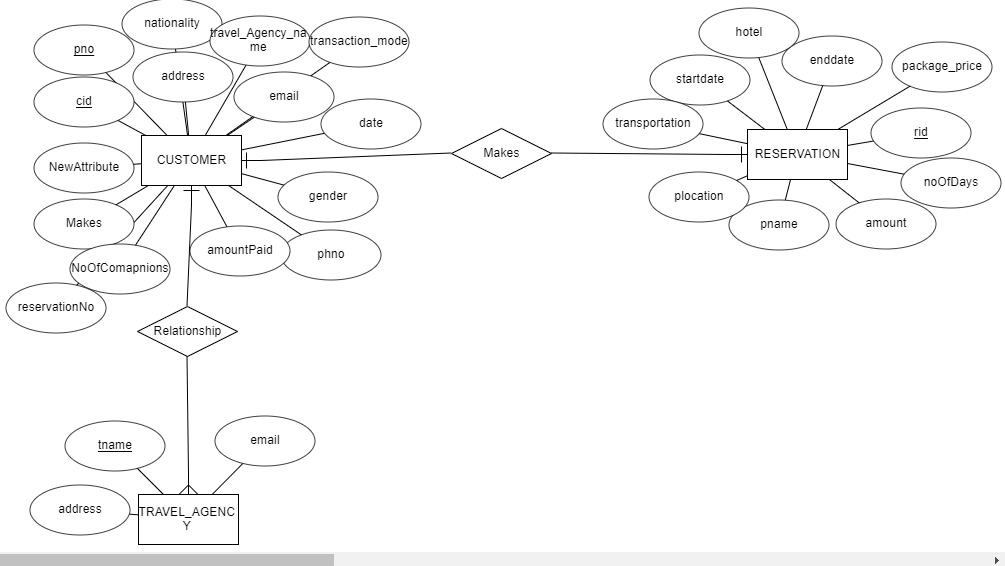
**TRAVEL \_AGECNY**

|  |  |  |
| --- | --- | --- |
| tname | Email | address |

**RESERVATION**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| rid | amount | package\_price | plocation | Hotel | plocation | transportation | noOfdays | start\_date | end\_date |

ER DIAGRAM



APPLYING 1NF :

**CUSTOMER**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | cid | pno | phone | address | nationality | gender | age | TransactionMode | tname | noofcompanion | Date | amountPaid | ReservationNo |

In CUSTOMER, all the values are atomic so it is in 1NF already.

**TRAVEL \_AGECNY**

|  |  |  |
| --- | --- | --- |
| tname | Email | address |

In TRAVEL\_AGENCY, all the attributes are atomic so it is in 1NF already.

**RESERVATION**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| rid | amount | package\_price | plocation | Hotel | plocation | transportation | noOfdays | start\_date | end\_date |

In Reservation, all the attributes are atomic so it is in 1NF already.

2NF:

**CUSTOMER**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cid | Pno | Phone | address | nationality | name | Gender | age | email | TransactionMode | tname | noofcomapnion | date | reservationno | amountpaid |

**PAYMENT\_MODE**

|  |  |  |
| --- | --- | --- |
| cid | Pno | transcactionmode |

**CUSTOMER**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | cid | phone | address | Nationality | gender | age | email | tname | noofcomapnion |

**PAYMENT**

|  |  |  |
| --- | --- | --- |
| Pno | price | date |

**TRAVEL\_AGENCY**

|  |  |  |
| --- | --- | --- |
| Tname | email | address |

**RESERVATION**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rid | amount | Package\_price | plocation | hotel | plocation | tranportation | noOfdays | startdate | endate |

3NF:

**CUSTOMER**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | cid | pno | Phone | address | nationality | gender | age | email | TransactionMode | tname | noofcomapnion | Date | ReservationNo | amountpaid |

Here there is a transitive dependency of non-key attribute on the primary key. Because non-key attribute amount\_paid depends on reservation\_no so we need to apply 3NF.After applying 3NF we get,

**CUSTOMER**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | cid | pno | Phone | Address | nationality | Gender | age | email | Transaction Mode | tname | noofcomapnion | Date | ReservationNo |

**RESERVATION\_AMOUNT**

|  |  |
| --- | --- |
| ReservationNo | amountpaid |

**PAYMENT\_MODE**

|  |  |  |
| --- | --- | --- |
| cid | Pno | transcactionmode |

Here there is no transitive dependency of non-key attribute on the primary key. So it is in 3NF.

**PAYMENT**

|  |  |  |
| --- | --- | --- |
| Pno | price | date |

Here there is no transitive dependency of non-key attribute on the primary key. So it is in 3NF.

**TRAVEL\_AGENCY**

|  |  |  |
| --- | --- | --- |
| tname | email | address |

Here there is no transitive dependency of non-key attribute on the primary key.So it is in 3NF.

**RESERVATION**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rid | amount | Package\_price | Pname | hotel | plocation | tranportation | noOfdays | startdate | endate |

Here there is a transitive dependency of non-key attribute on the primary key. Because non-key attributes hotel, package\_price,plocation,transportation,no\_of\_days,start\_date,end\_date depends on non-key attribute pname. Hence we should apply 3NF.So after applying 3NF we get,

**RESERVATION**

|  |  |  |
| --- | --- | --- |
| rid | amount | pno |

**PACKAGES**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Package\_price | Pname | hotel | plocation | tranportation | noOfdays | startdate | endate |

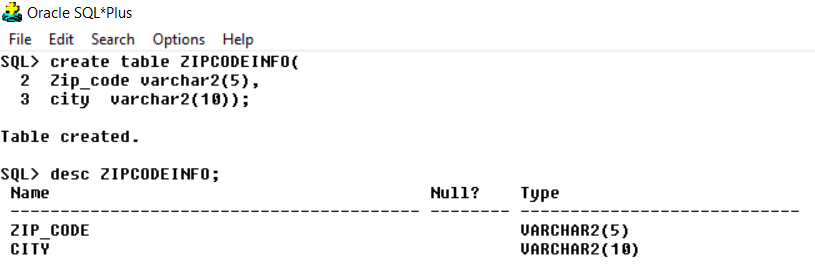
**NAME: Deepti Nayak**

**USN: 2SD18IS016**

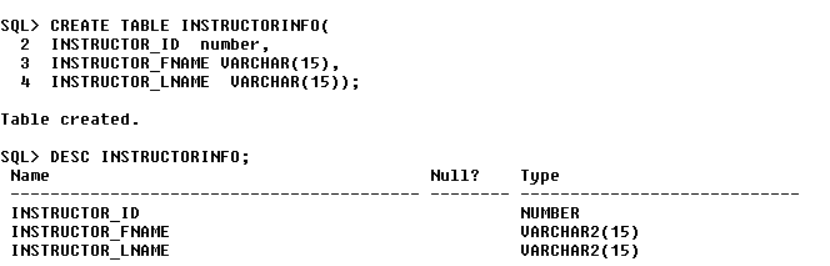
**SUBJECT:** DBMS PROJECT

**CREATION OF TABLES**

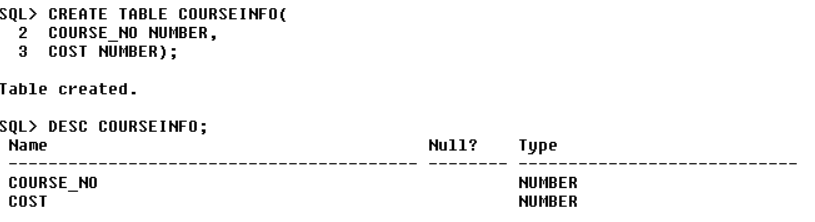
**1] TABLE NAME:** ZIPCODEINFO



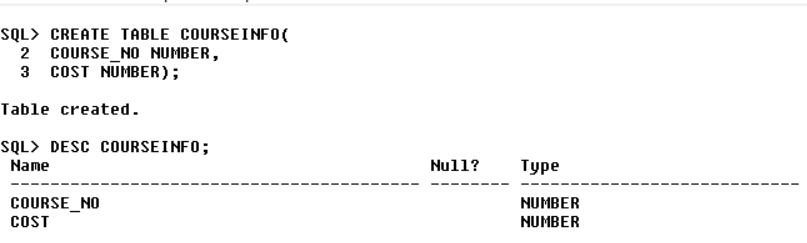
**2]** **TABLE NAME:** INSTRUCTORINFO



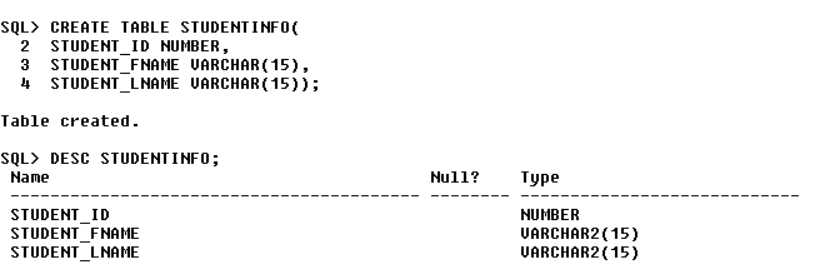
**3]** **TABLE NAME:** COURSEINFO



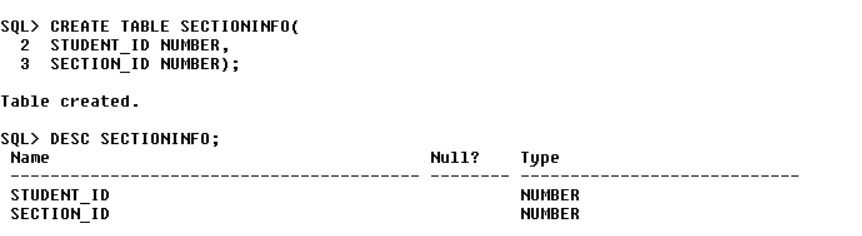
**4] TABLE NAME:** COURSEINFO

****

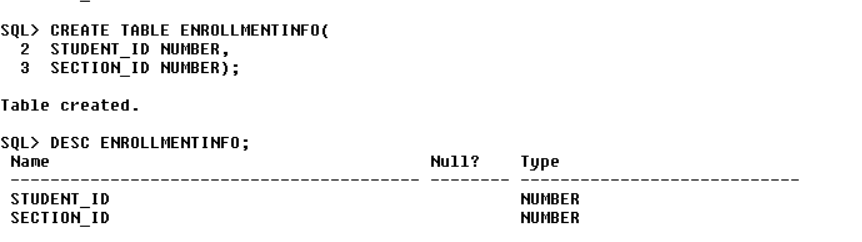
**5] TABLE NAME:** STUDENTINFO

****

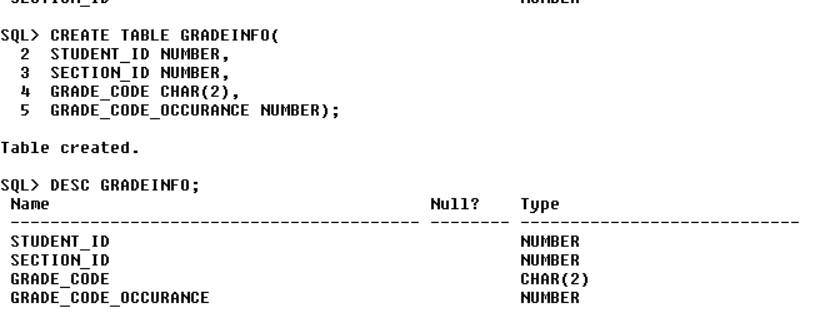
**6] TABLE NAME:** SECTIONINFO

****

**7] TABLE NAME:** ENROLLMENTINFO

****

**8] TABLE NAME:** GRADEINFO

****

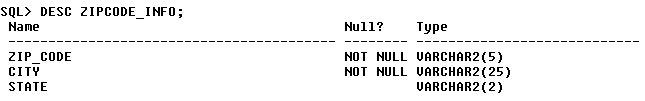
**ALTERING TABLES**

**1] QUERIES:**

* ALTER TABLE ZIPCODEINFO RENAME TO ZIPCODE\_INFO;
* ALTER TABLE ZIPCODE\_INFO ADD STATE VARCHAR2(15);

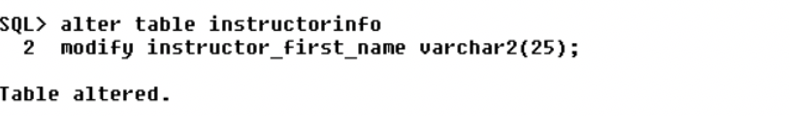


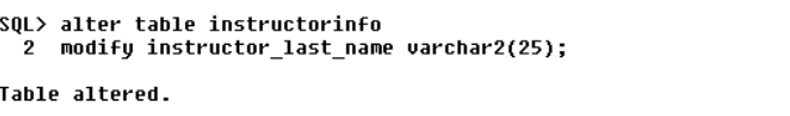
**OUTPUT:**

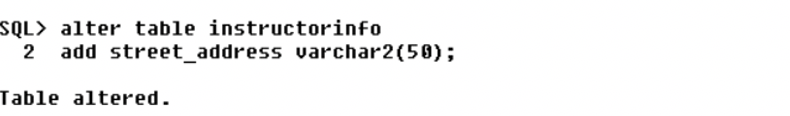


**2] QUERIES:**

* ALTER TABLE INSTRUCTORINFO RENAME TO INSTRUCTOR\_INFO;

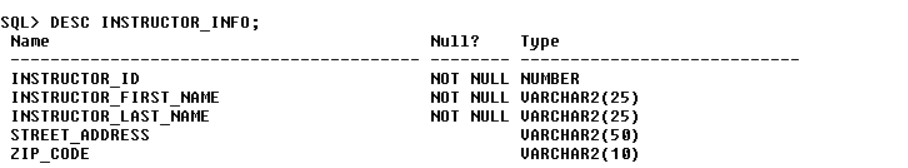
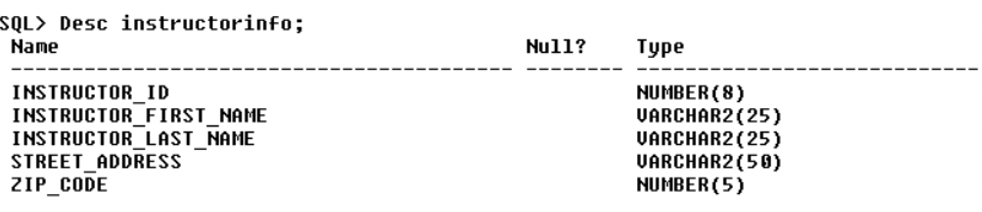








**OUTPUT:**

****

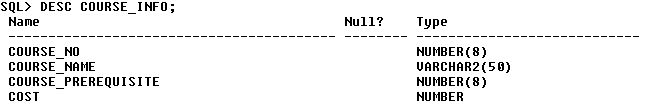
**3] QUERIES:**

* ALTER TABLE COURSEINFO RENAME TO COURSE\_INFO;





**OUTPUT:**



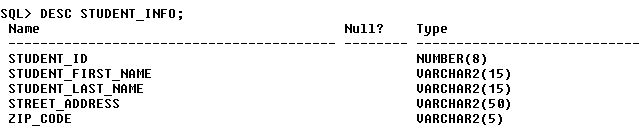
**4] QUERIES:**

* ALTER TABLE STUDENTINFO RENAME TO STUDENT\_INFO;
* ALTER TABLE STUDENT\_INFO RENAME COLUMN STUDENT\_FNAME TO STUDENT\_FIRST\_NAME;
* ALTER TABLE STUDENT\_INFO RENAME COLUMN STUDENT\_LNAME TO STUDENT\_LAST\_NAME;



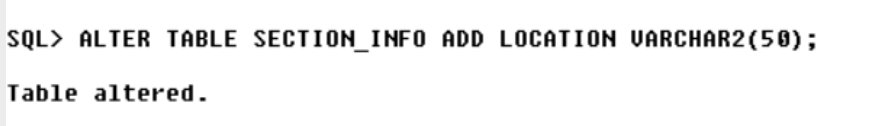


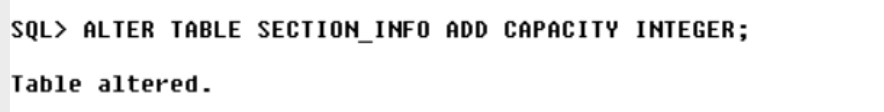
**OUTPUT:**



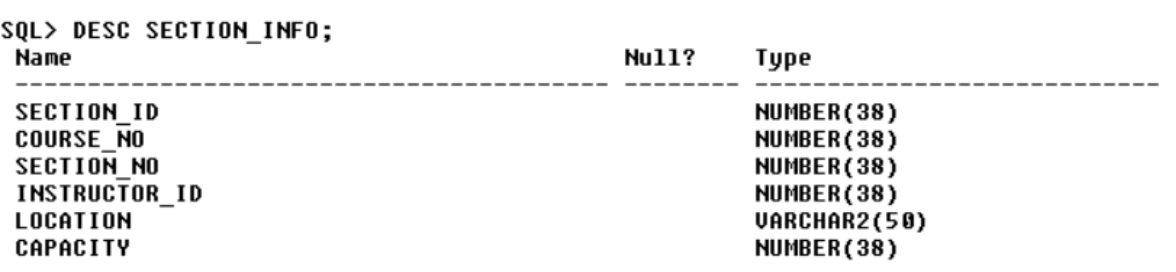
**5] QUERIES:**

* ALTER TABLE SECTIONINFO RENAME TO SECTION\_INFO;
* ALTER TABLE SECTION\_INFO RENAME COLUMN STUDENT\_ID TO SECTION\_NO;
* ALTER TABLE SECTION\_INFO ADD COURSE\_NO NUMBER;
* ALTER TABLE SECTION\_INFO ADD INSTRUCTOR\_ID NUMBER;





**OUTPUT:**

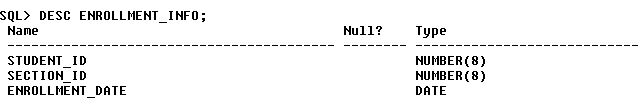
****

**6] QUERIES:**

* ALTER TABLE ENROLLMENTINFO RENAME TO ENROLLMENT\_INFO;



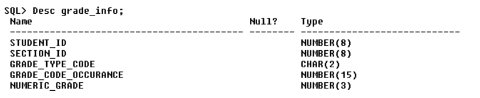
**OUPUT:**



**7] QUERIES:**

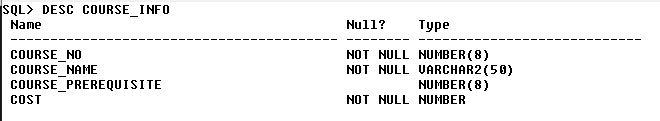
* ALTER TABLE GRADE\_INFO RENAME COLUMN GRADE\_CODE TO GRADE\_TYPE\_CODE;
* ALTER TABLE GRADE\_INFO ADD NUMERIC GRADE NUMBER(8);

**OUTPUT:**

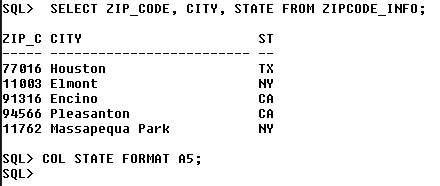


**SIMPLE SQL QUERIES**

1. Display the structure of the course table.



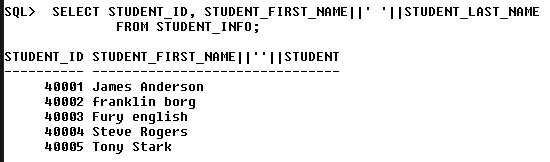
2. Display the zipcode, city and state. Observe the column heading of state column. If not appearing correctly, give a proper heading.



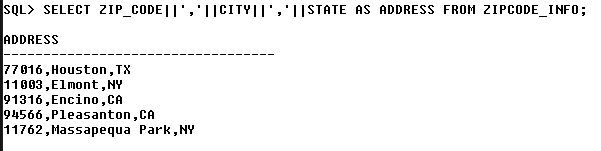
3. Display the unique states



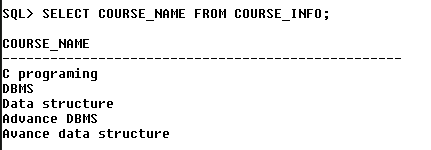
4. Display the student\_id, name. Concatenate the first\_name and last name.



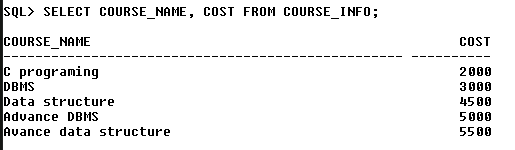
5. Display the Zipcode, city and state as a single column. Separate the data with a comma. E.g 400050, Mumbai, MH. Give the column heading as Address.



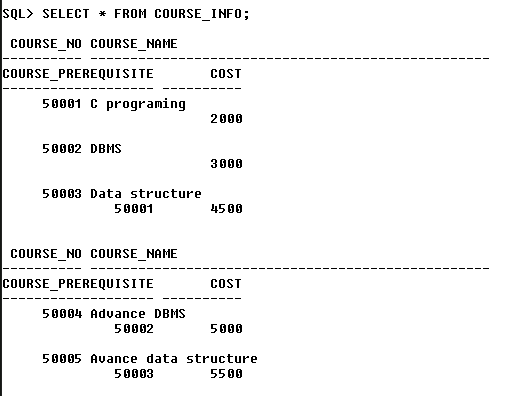
6. Display the description of the course



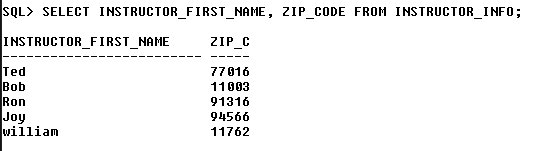
7. In the above query also display the cost.



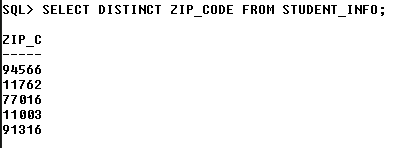
8. Display all the columns of the course\_info table.



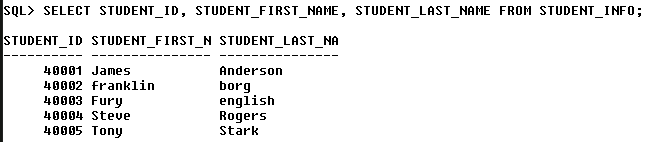
9. Display the instructor’s last name and the zip in which the instructor resides.



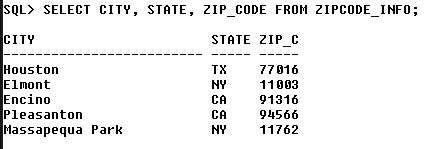
10. There are many students who are living in the area, display unique zip’s only.



11. Write a select statement to list the first and last names of all students.

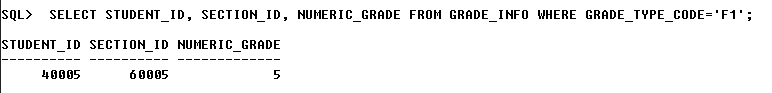


12. Write a select statement to list all cities, their states and zip codes.

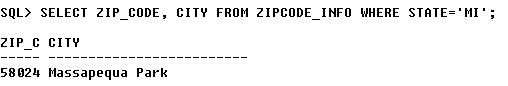


**SELECT USING WHERE**

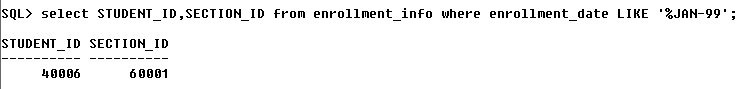
1.Display the studenUd, section\_id and numeric\_grade of those students who have the grade code type as ‘Fl’



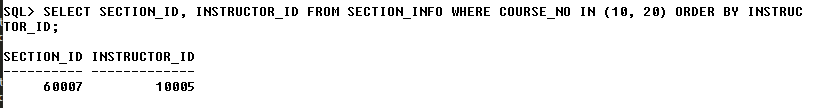
2. Display the Zipcode and cities for the state ‘MI’



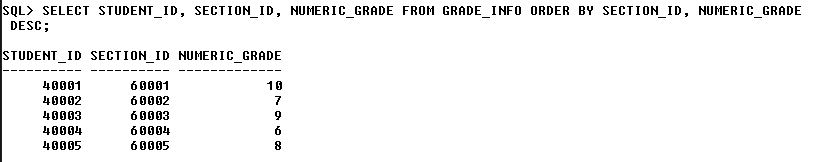
3. Display the student details that have enrolled in the month of Jan 1999. Sort the data in the ascending order of student names. Note: Names should be concatenated.



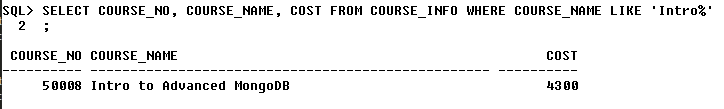
4. Display the section and the instructor id of the course 10 and 20. Sort the data in the ascending order of instructor id.



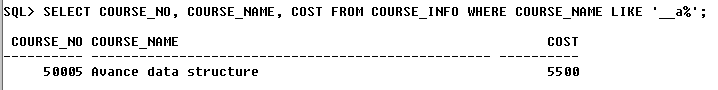
5. Display the studenUd, section\_id and numeric\_grade. Sort the section\_id in ascending order followed by numeric\_grade in descending order.



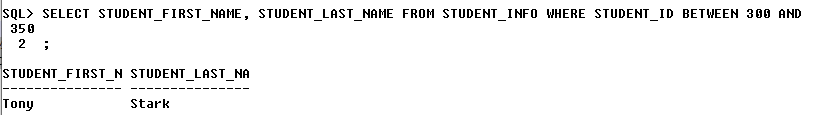
6. Display the course\_no, description and cost of the courses. The courses should have the word ‘Intro’



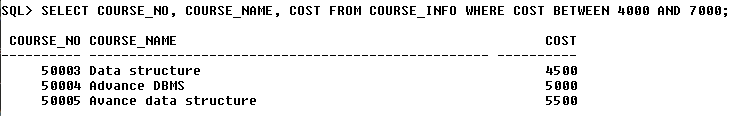
7. Display the course details whose third from last letter is ‘a’



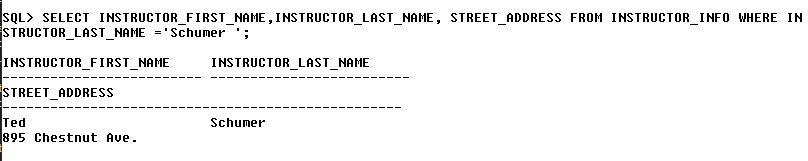
8. Display the student names whose studenUd is in the range b of 300 to 350.



9. Display the course details whose cost ranges from 4000 to 7000.

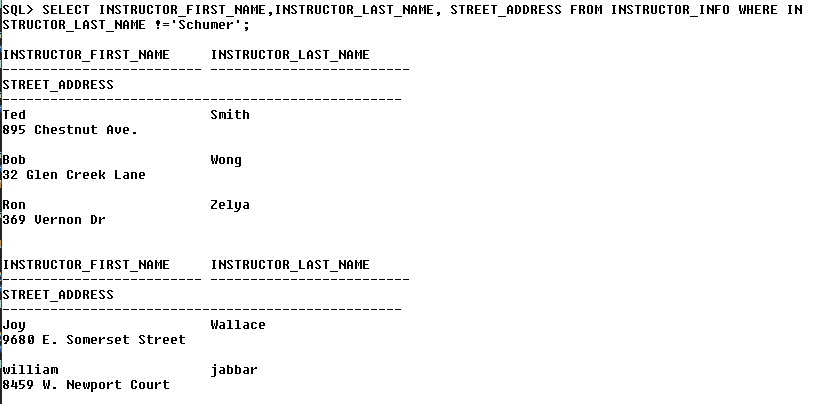


10. Write a select statement that displays the instructor&#39;s first name whose last name is ‘Schumer’

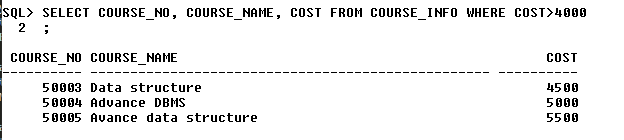


11. Write a select statement that displays the instructor’s first name whose last

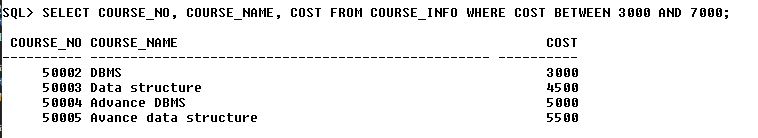
name is not ‘Schumer’



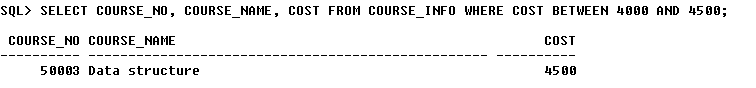
12. Display the course name and cost of those courses whose cost is more than 4000.



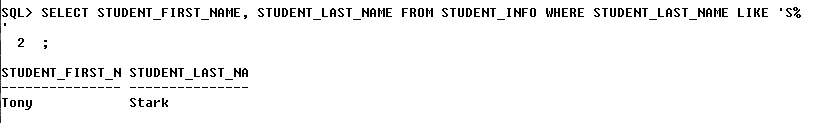
13. Display the course name and cost of those courses whose cost is in the range of 3000 and 7000.



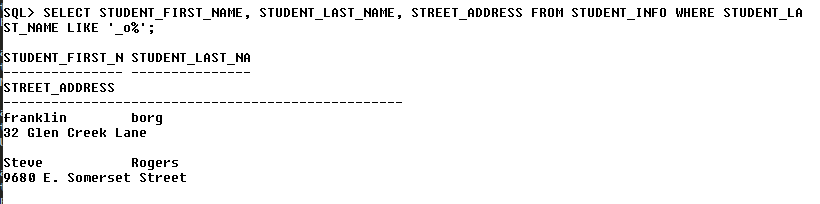
14. Display the course name and cost of those courses whose cost is 4000 and 4500.



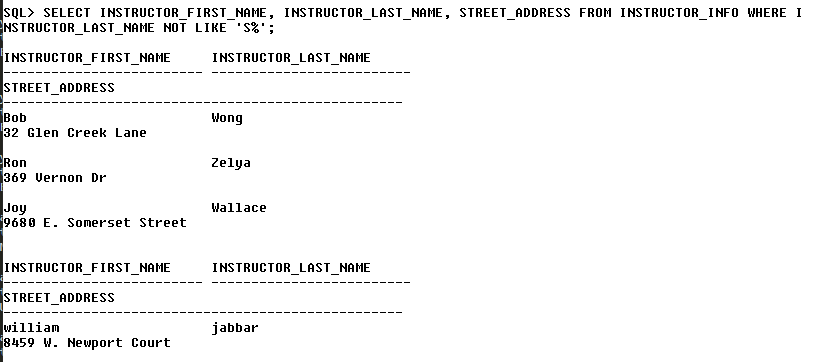
15. Write a select statement that displays the student’s first name and address whose last name starts with ‘S’.



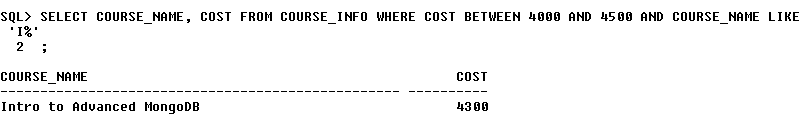
16. Write a select statement that displays the student&#39;s first name and address whose last name contains ‘0’ as the second letter.



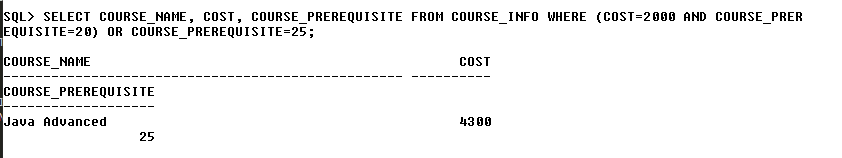
17. Write a select statement that displays the instructor’s first name whose last name does not start with ’S’



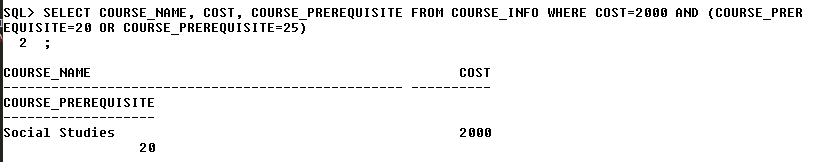
18. Display the course name and cost of those courses whose cost is in range of 4000 and 4500 and the course name starts with ‘I’.



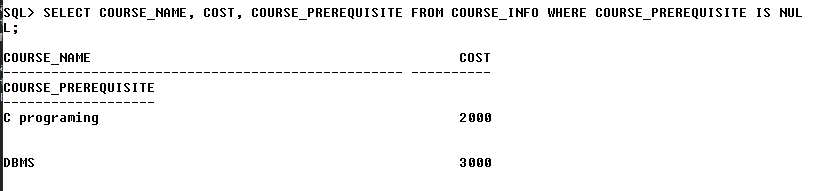
19. Display the course name and cost and prerequisite of those courses whose cost is 2000 and prerequisite is 20 and all courses whose prerequisite is 25.



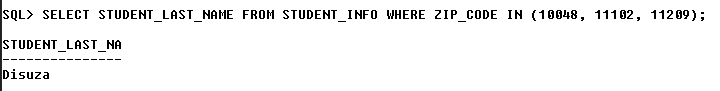
20. Display the course name and cost and prerequisite of those courses whose cost is 2000 and all records whose prerequisite is 20 and 25.



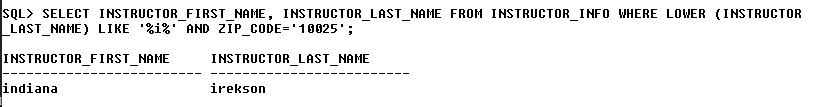
21. Display the records of the course\_info table which have no prerequisite.



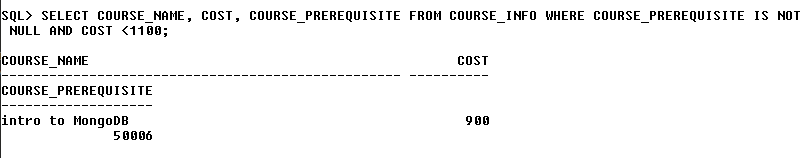
22. Write a select statement to list the last names of students living either in zip code 10048, 11102 and 11209.



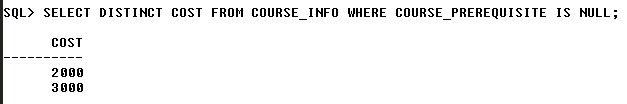
23. Write a select statement to list the first and last names of instructors with the letter ‘I’ in any case in the last name and living in zip code 10025.



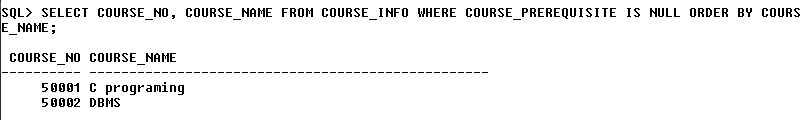
24. Write a select statement to list descriptions of courses which have prerequisites and cost is less than 1100.



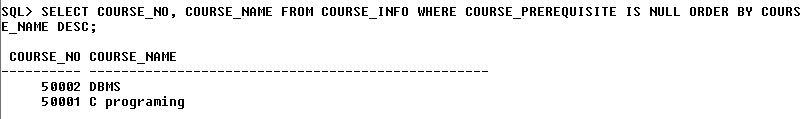
25. Write a select statement to list the cost of courses that do not a prerequisite. In the result the cost should not be repeated.



26. Display the course\_no and the coursename who do not a prerequisite. Sort the data on the basis of coursename.



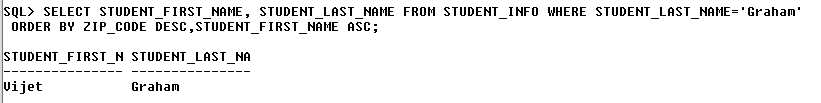
27. For the above query sort in the descending order of coursename.



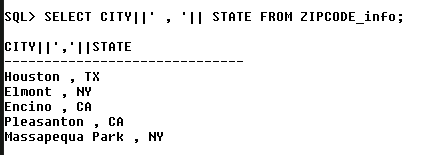
28. Write a select statement to list each city in state New York or Connecticut sorted in ascending order by zipcode.



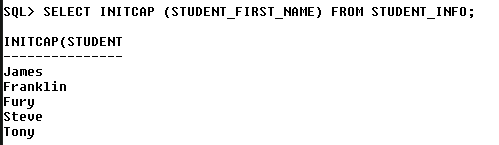
29. Display the zip, first and last name of students with the last name Graham. Order the result by zip in descending order and the first name in ascending order.



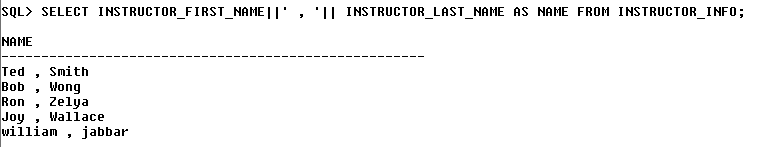
30. Display the city and state as 1 column.



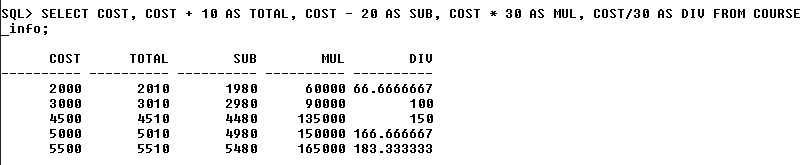
31. Display the first name of the student. Only the first letter of the first name should be in uppercase.



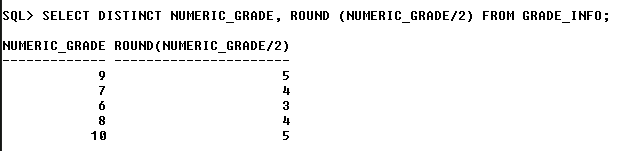
32. Write a select statement that displays the instructor&#39;s first name followed by comma and a space which is followed by last\_name. The result should be displayed in 1 result set.



33. Display the select statement that displays the cost, add 10 to cost, subtract 20 from cost, multiply cost by 30 and divide cost by 5.



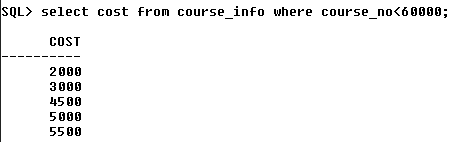
34. Write a select statement that displays unique numeric grade, half these values and do not display this value in decimals. This calculation should be displayed as a separate column.



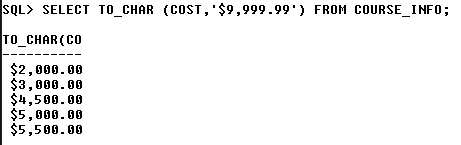
**SELECT WITH TO\_CHAR/DECODE/IS NULL**

1. Display the cost in the following format for all the courses whose course\_no is less

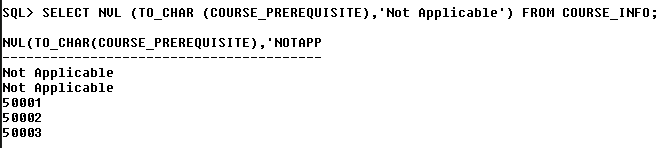
than 25.

****

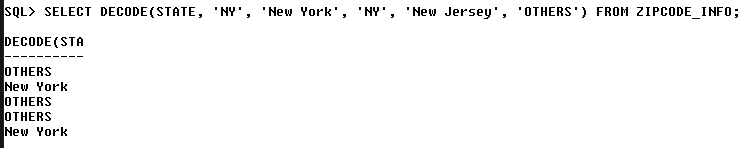
2. Write a query to format the cost column. The cost displayed should have a leading $ sign followed by a comma to separate the thousands and should display 2 decimals.



3. For those courses having no pre-requisites display data as ‘Not applicable’

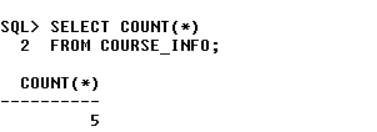


4. Display the state as ‘New York’ for NY, ‘New Jersey’ for NY and ‘Others’ for any other state.

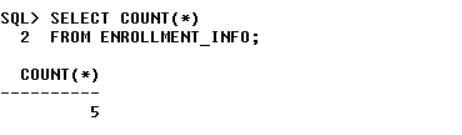


**AGGREGATE FUNCTIONS**

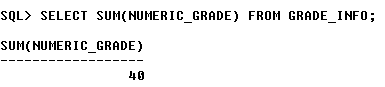
1.Display the count of records in the course table



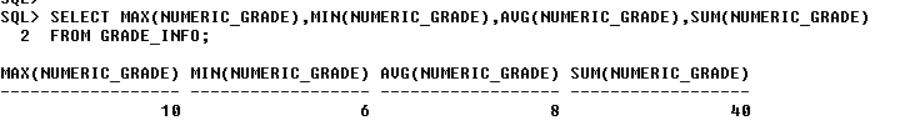
2. Display the total number of records in enrollment table .



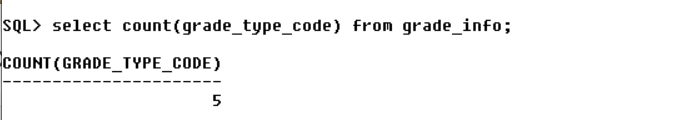
3. Display sum of numeric\_grade from grade\_info .



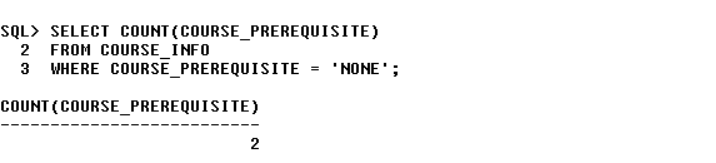
4. Display the average, total, minimum and maximum numeric grade.



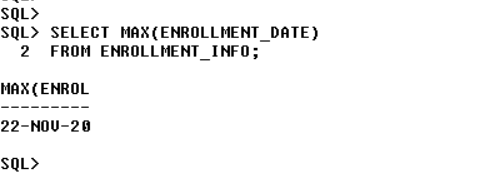
5. Display the count of grade code type.



6. Write a SELECT statement that displays the total number of courses which do not have any pre- requisite.

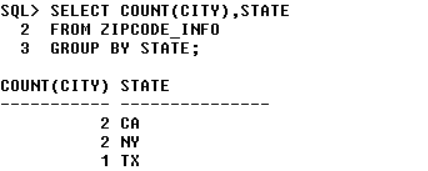


7. Display the date of the student who got recently enrolled.

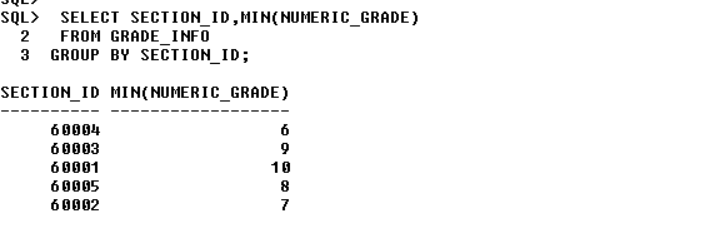


**GROUP BY AND HAVING CLAUSE**

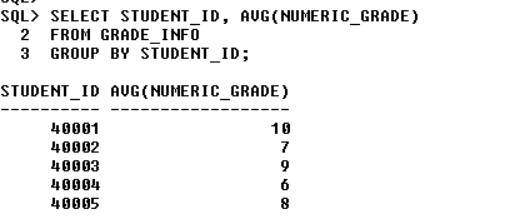
1.Display the count of cities for each state.



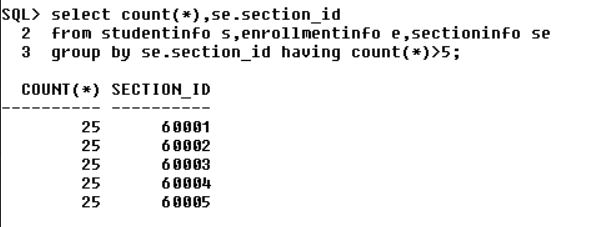
2. Display the minimum numeric grade section wise for each student.



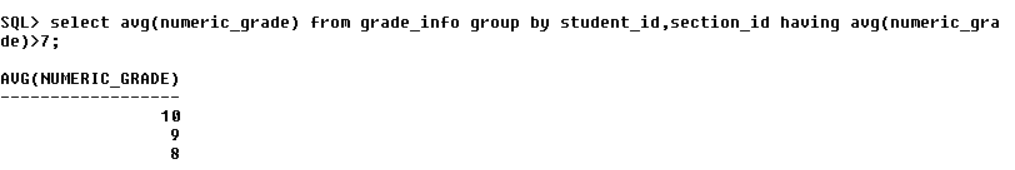
3. Display the average numeric grade for each student.



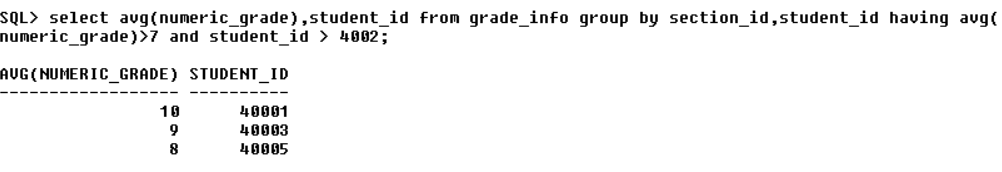
4. Display the count of students enrolled in each section. Display only those sections where the number of students enrolled is more than 5.



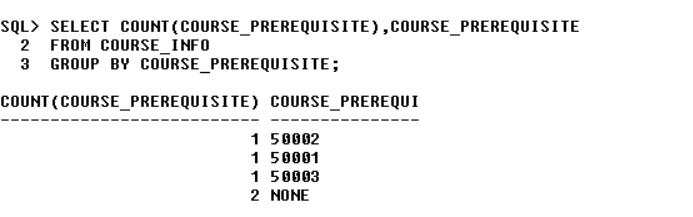
5. Display the average numeric grade for each student and section. The average numeric grade should be more than 75.



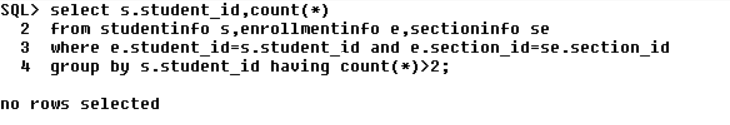
6. For the above query display the data for student\_id more than 280



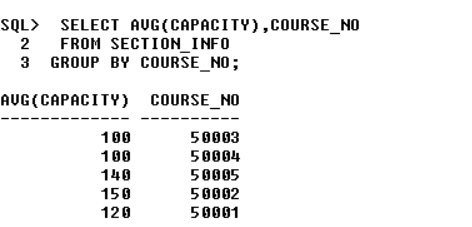
7. Display each prerequisite and its count from the course\_info table.



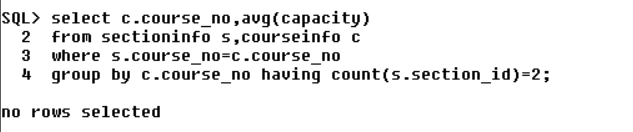
8. Display student\_id and number of courses they are enrolled in. Those students who are enrolled in more than 2 courses should be displayed.



9. Display the average capacity of each course.



10. For the above query display those courses which have exactly 2 sections.

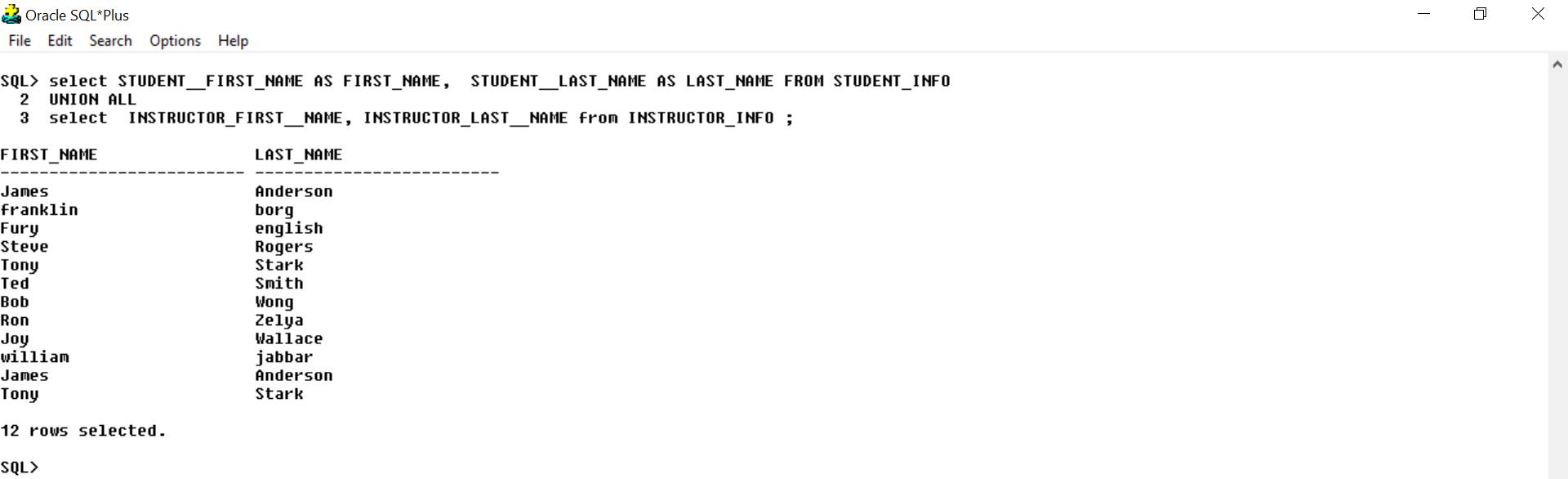


**UNION/INTERSECT AND MINUS**

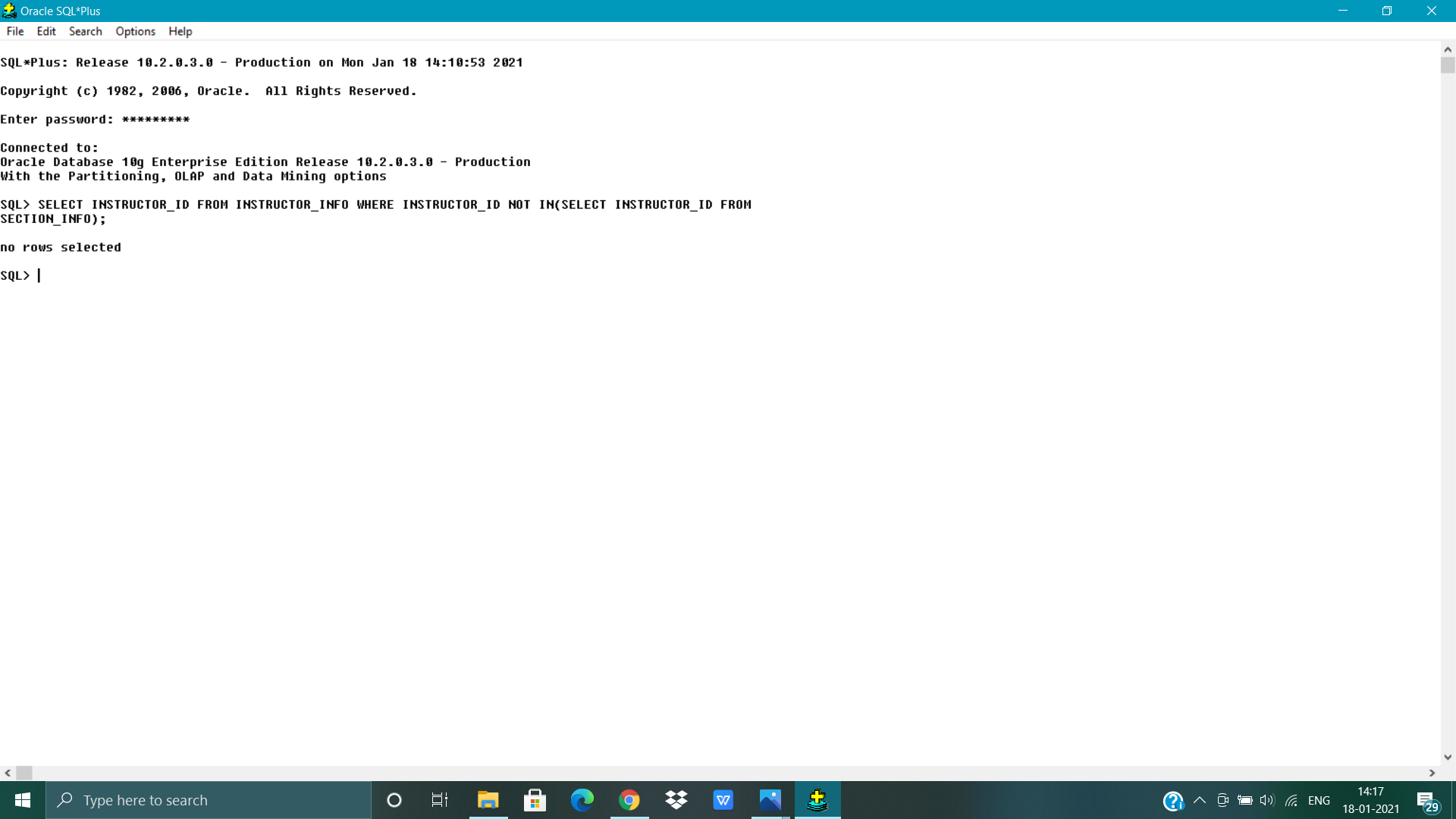
1.Display the first name and last name of both the instructor and student in one result set.



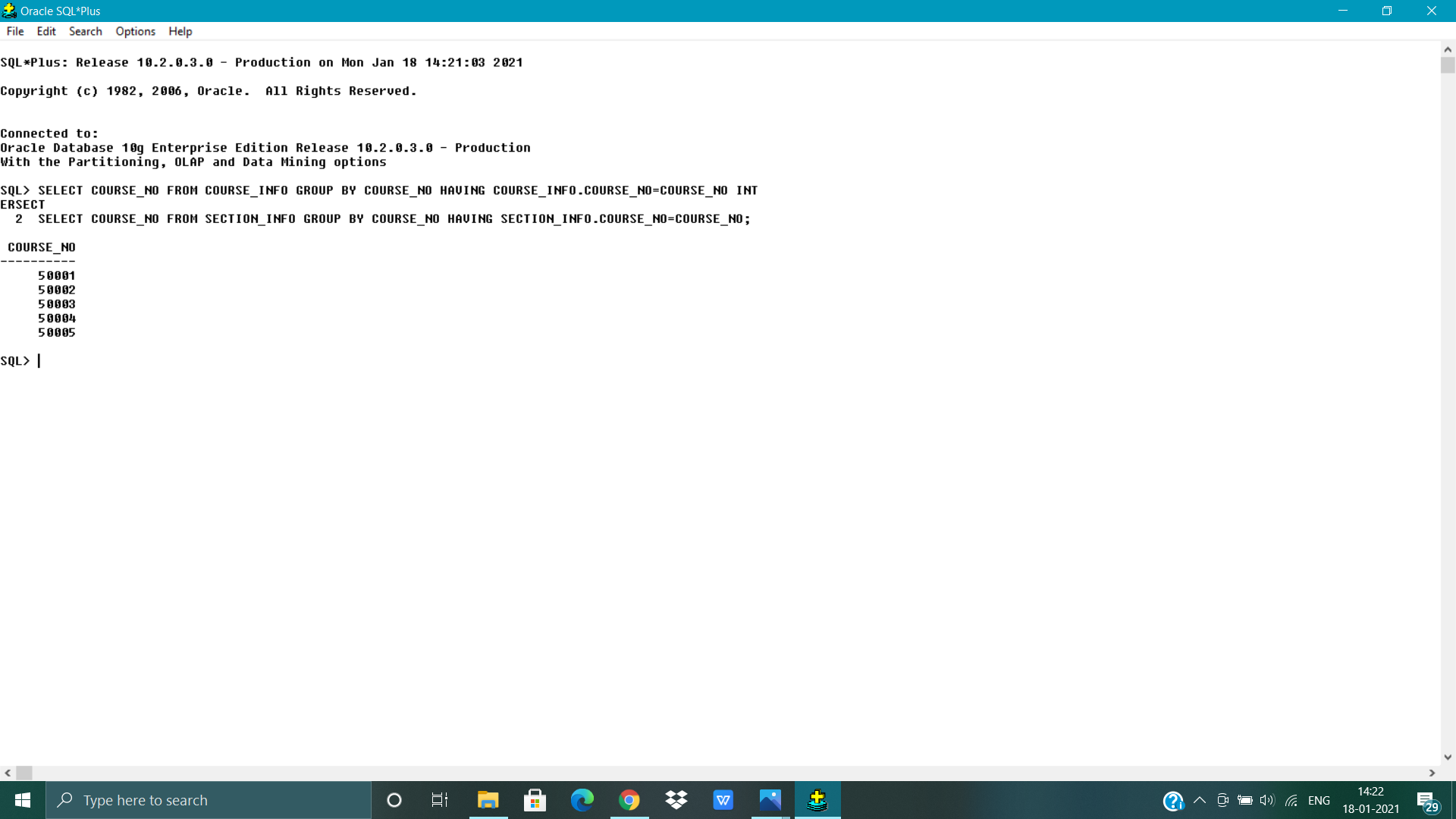
2. Modify the above query to display duplicate names also.



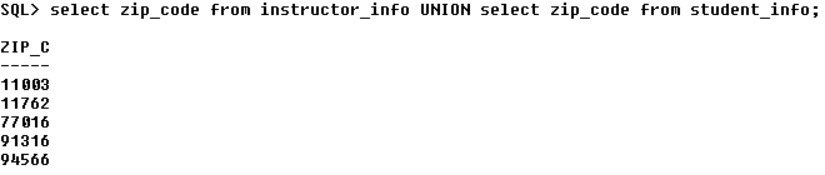
4. Display the instructor id not having any section.



5. Display the course no having section.



7. Display all the zip codes that are present in both the instructor and the student tables.



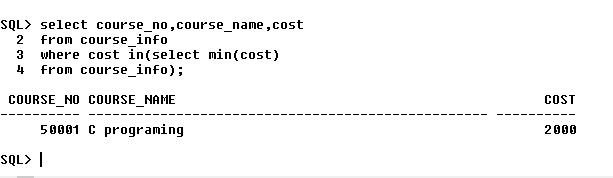
8. Display the student ids who have enrolled.



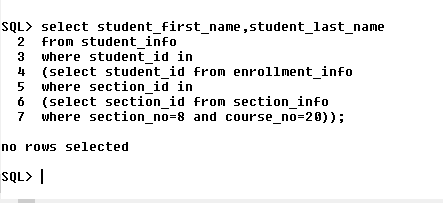
**SUBQUERIES**

**Subqueries returning 1 row**

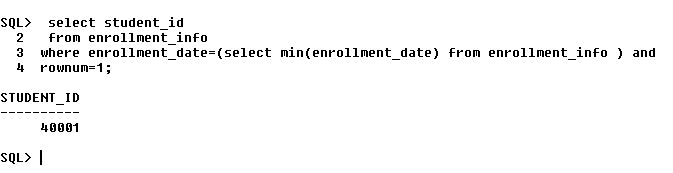
1.Display the course\_no, description and cost. Those records should be displayed whose cost is equal to the minimum cost.

****

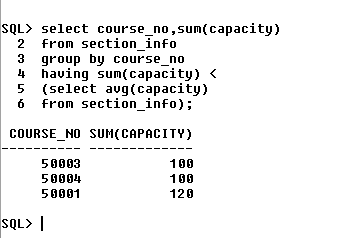
2. Display the name of the students enrolled in section no 8 and course number 20.

****

3. Display the student id’s who registered first.

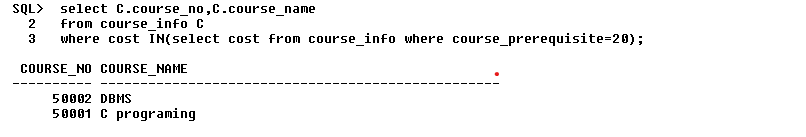
****

4. Display the course\_no and sum of capacity where the total capacity is less than the average capacity.

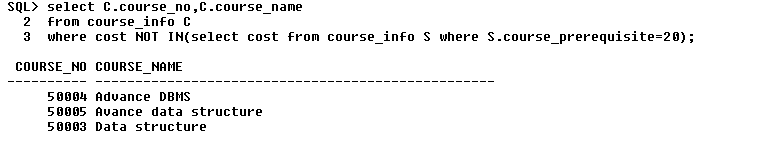
****

**Subqueries returning multiple rows**

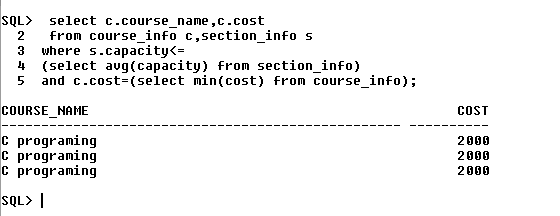
1. Display the course\_no and course\_name whose cost is same as that of course whose prerequisite 20



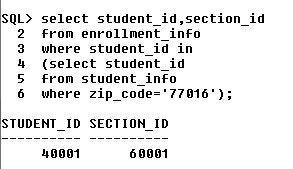
2. Modify the above query to display the records whose cost is not same as that of the course whose prerequisite is 20.



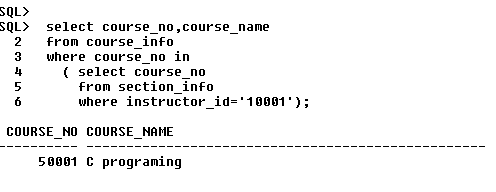
3. Display the course description and cost of the courses where the capacity is less than or equal to the average capacity and cost is equal to minimum cost.



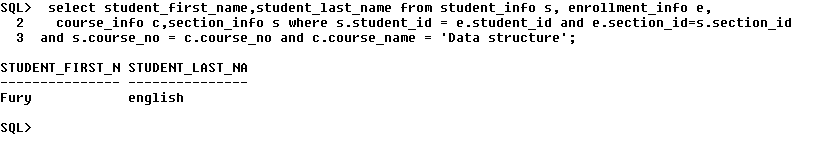
4. Display the student id and section id who are living in zip 77016.



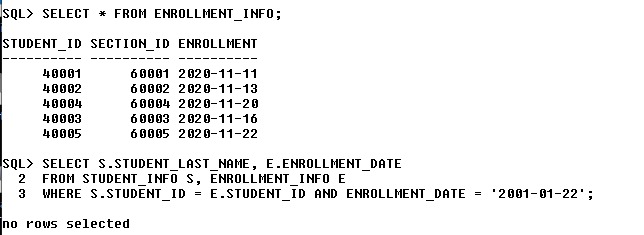
5. Display the course no and course\_name taught by instructor whose instructor ID =10001.



6. Display the students enrolled in course ‘Introduction to java’.

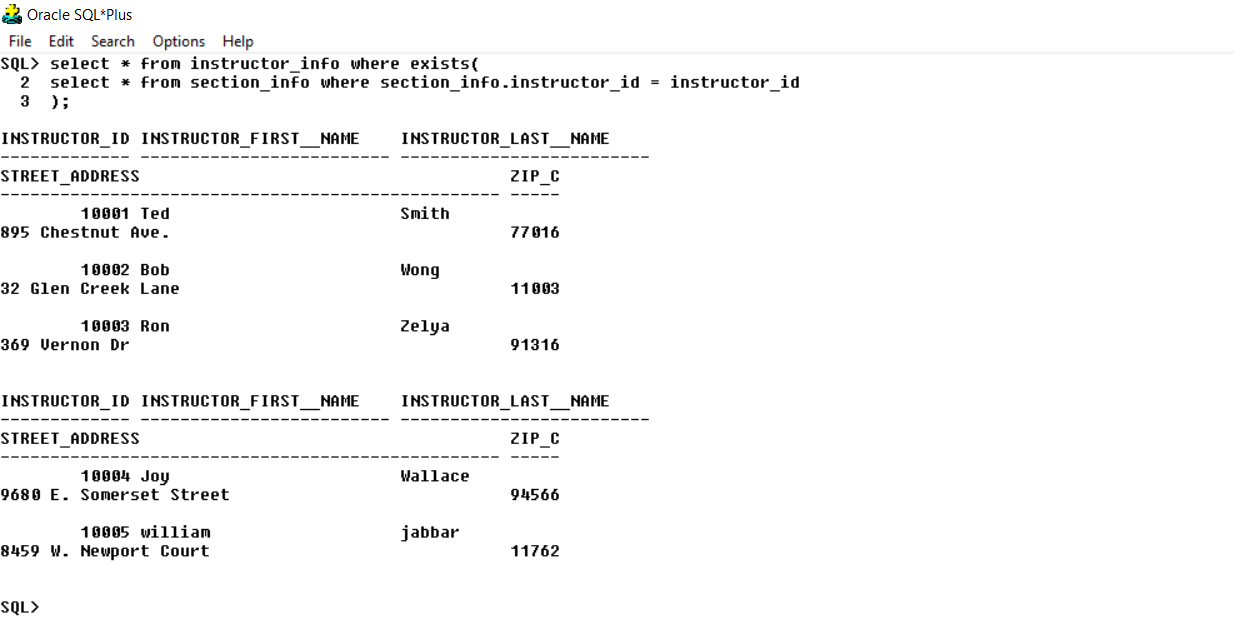


7. Display the last\_name and enrollment\_date from student\_info of those students who have enrolled on 22nd of Jan 2001.

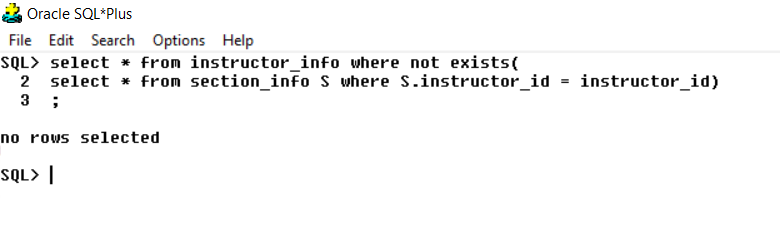


**Co-related subqueries with exists**

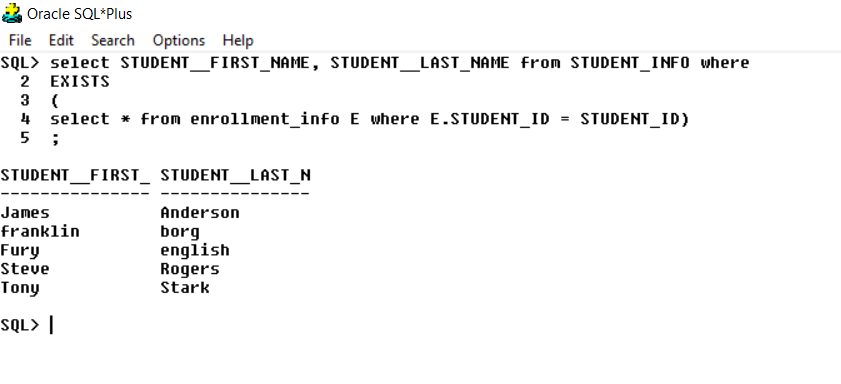
2. Display the instructor details only if the instructor teaches a section.

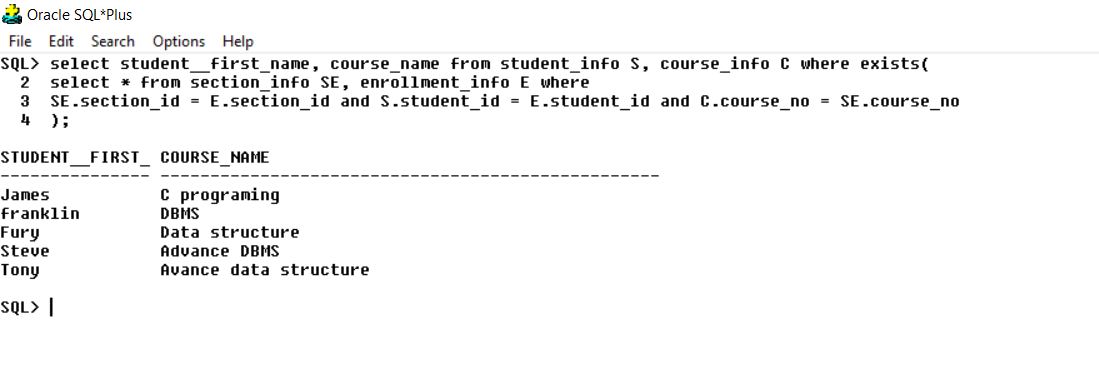


3. Display the details of those instructors who do not teach a section.

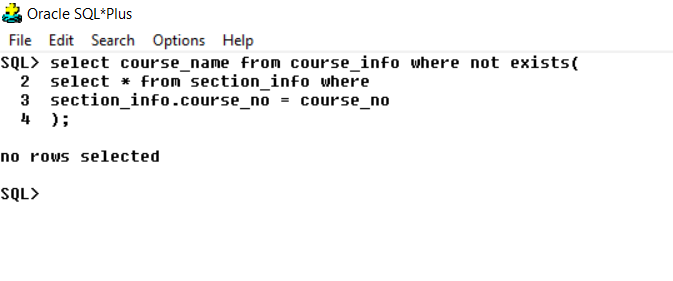


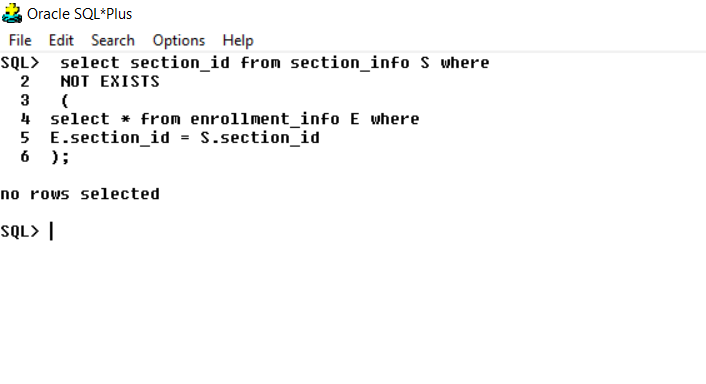
4. Display the names of the students who are enrolled.



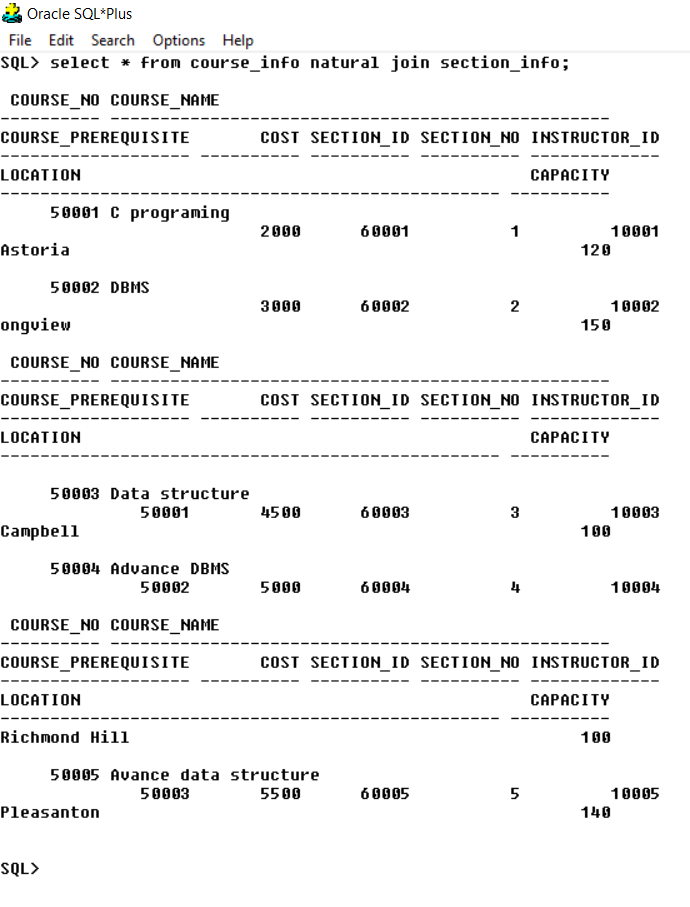
5. Display the courses enrolled by the students. 

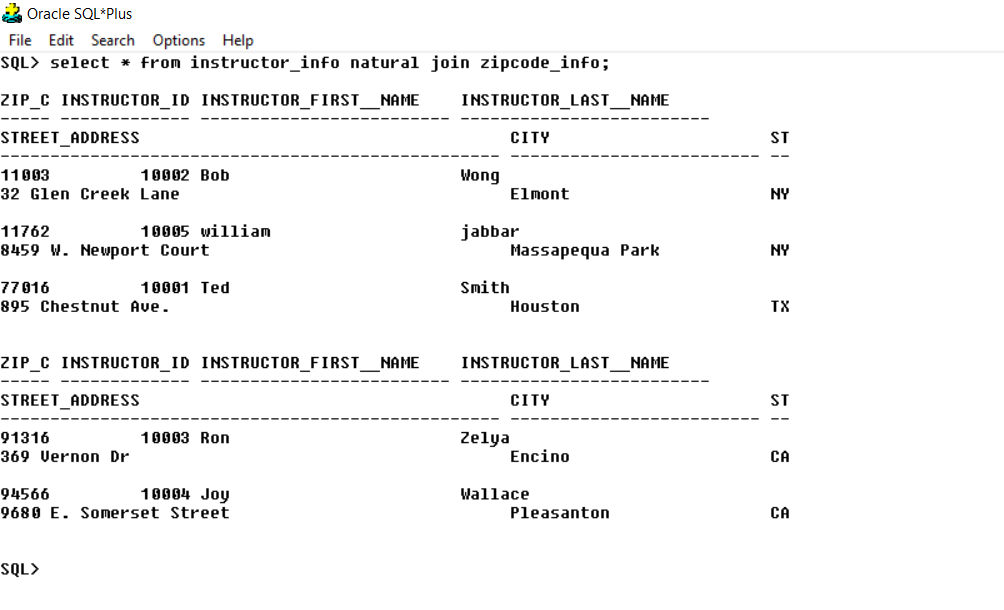
6. Display the courses who do not have sections.



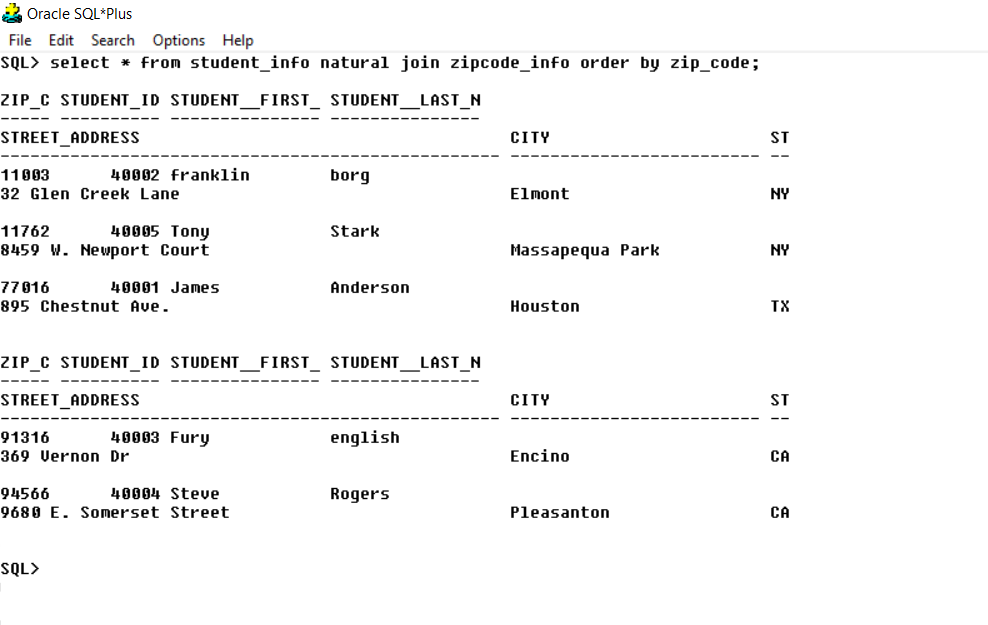
7. Display the sections in which no student is enrolled. 

**JOINS**

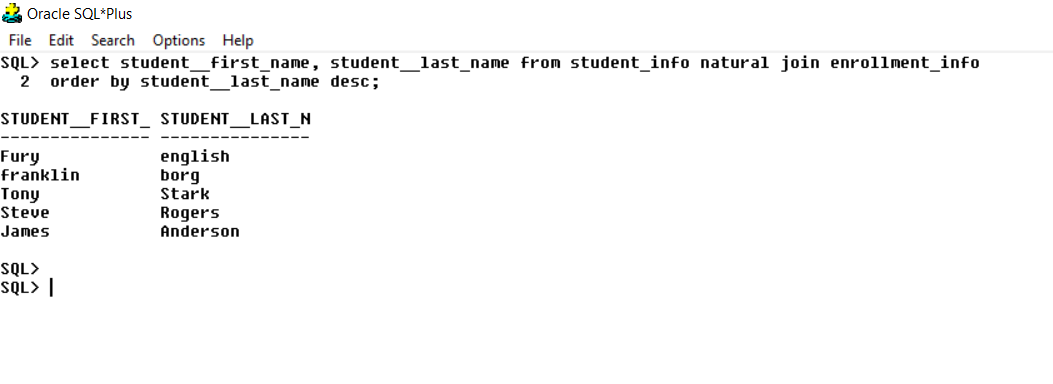
1. Display the course and the section information. 

2. Display the instructor details along with the city and the state that they live in. 

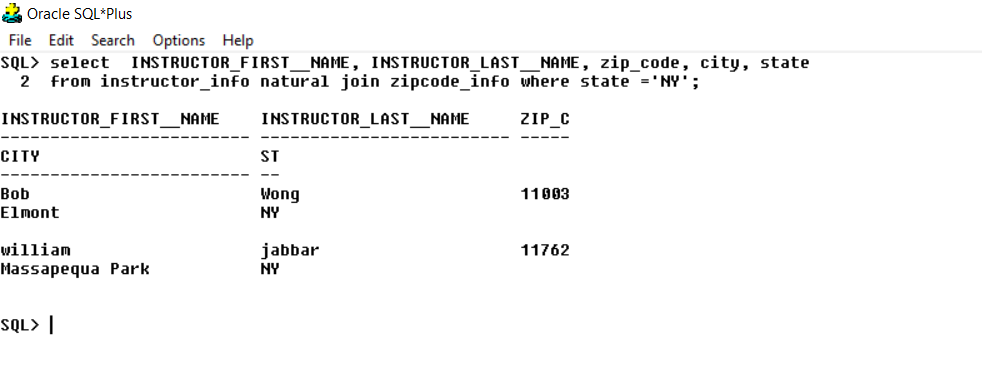
3. Display the student details of all students and city and state. Sort the result on the basis of zip code.



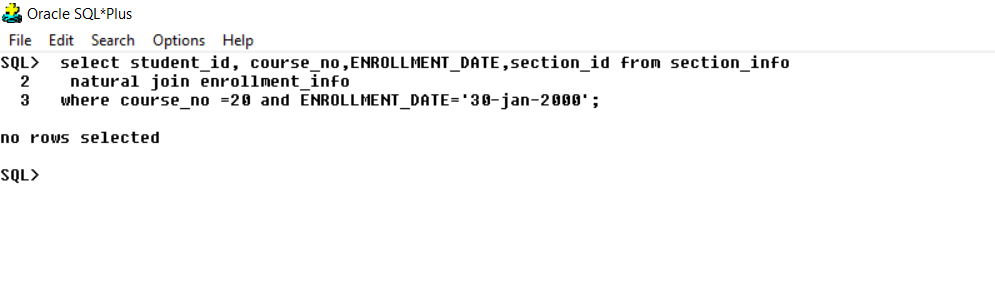
4. Write a query to display the first and last names of only the enrolled students and order on the basis of last name in descending order.

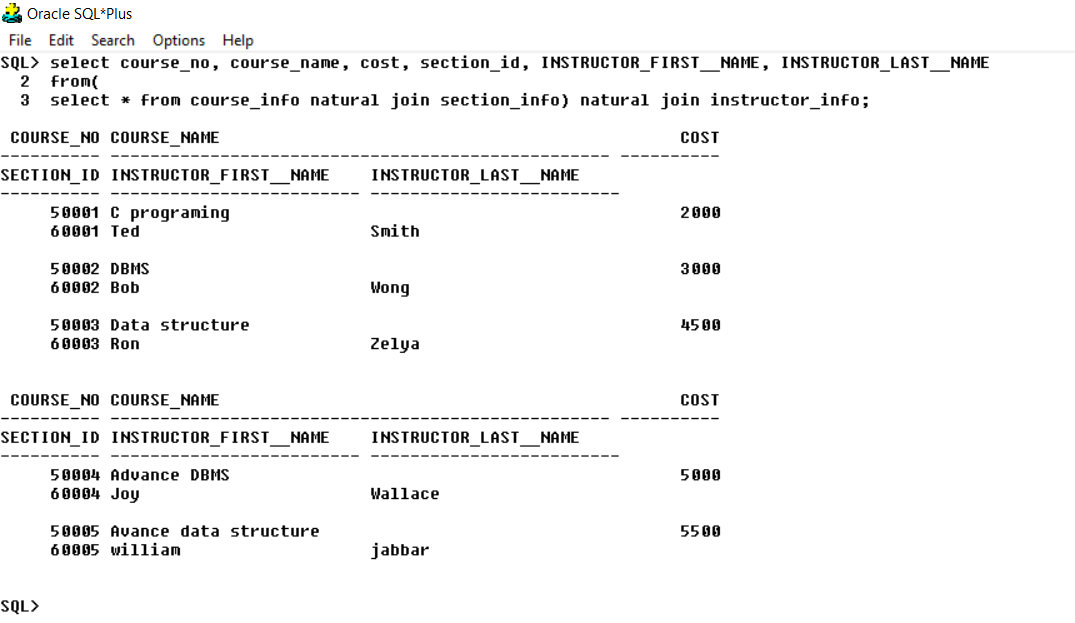


5. Display the zip, city and state and instructor names who are living in state ‘NY’.

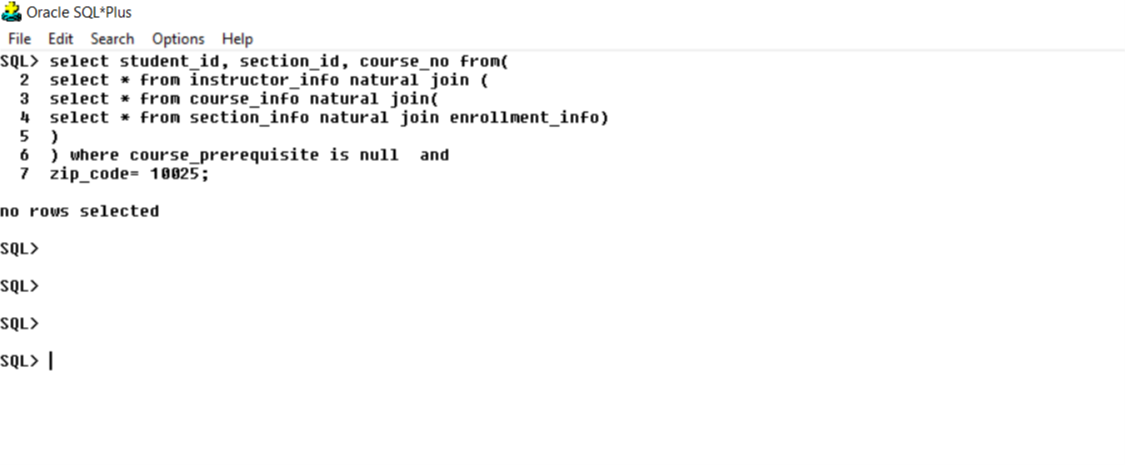


6. Write a query to display the student number, course number, enrollment date and section id for students who are enrolled in course number 20 on January 30 2000.

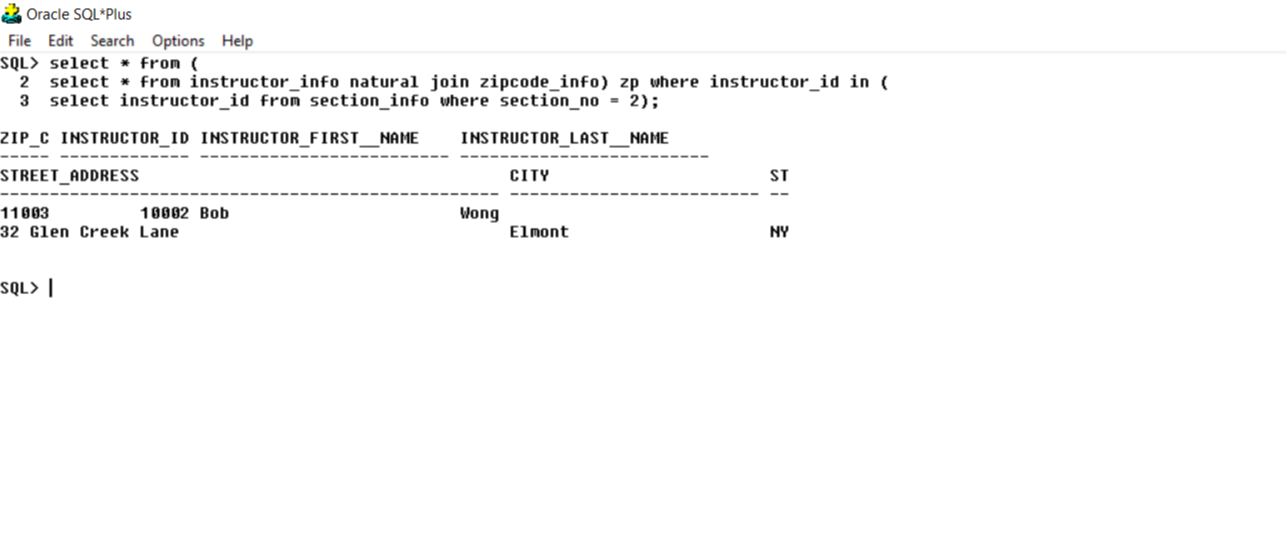


7. Display the course number, description, cost, section id, and the instructor names who are teaching the course. 

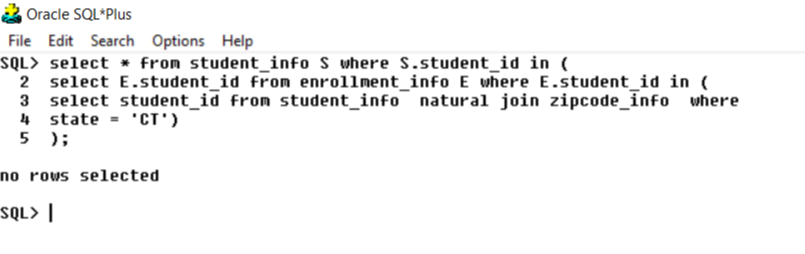
9. Display the student id, course number and section number of enrolled students where the instructor of the section lives in zip code 10025. Also, the course should not have any pre-requisites.



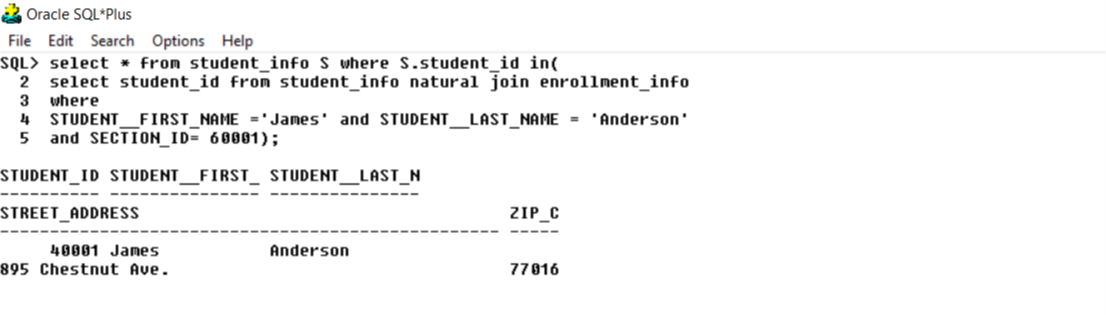
10. Display the instructor details along with the city, state who are teaching the section 2.

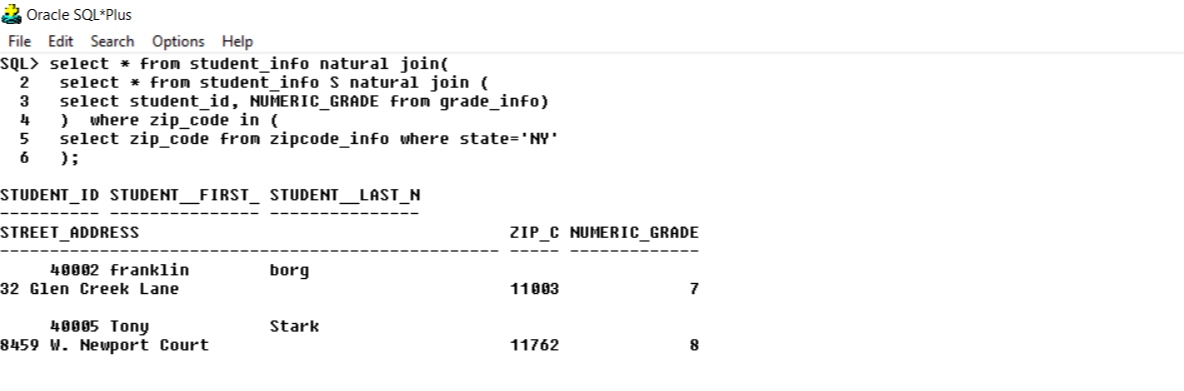


11. Display the list of enrolled student numbers who are living in state ‘CT’.

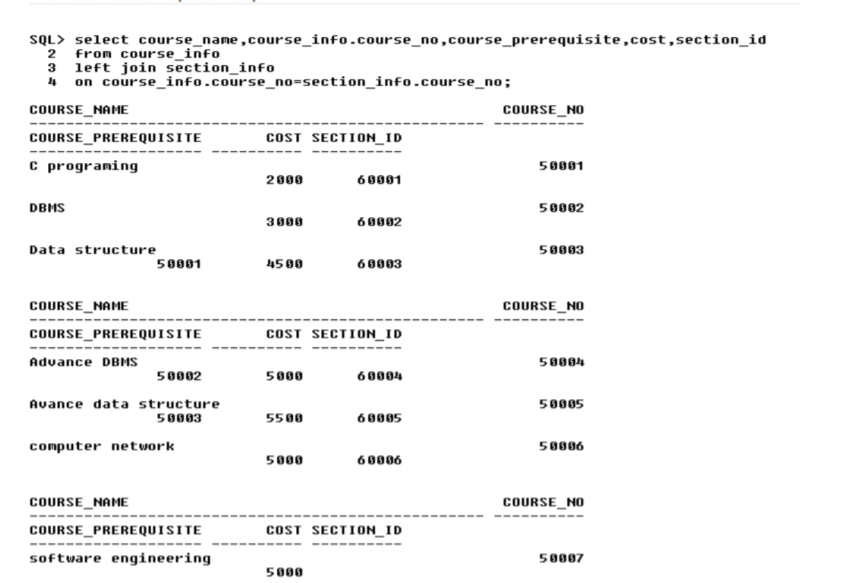


12. Write a query to display the details of student ‘ James Anderson' for section

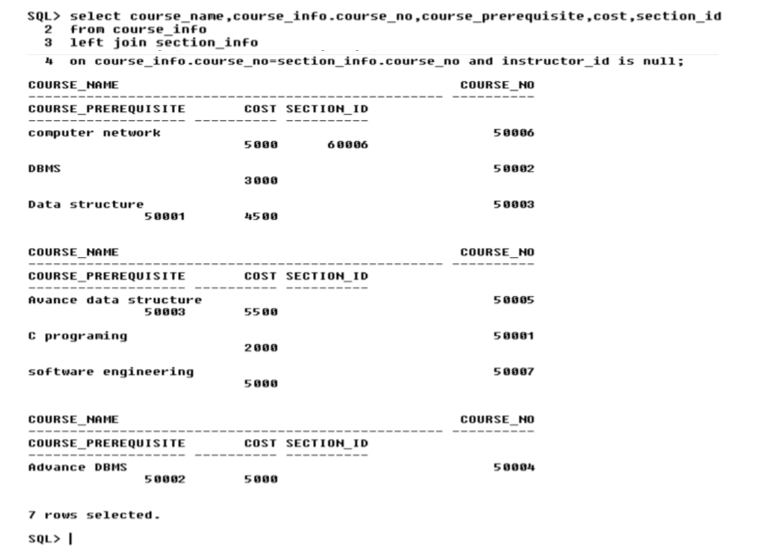
id 60001.

13. Display the final examination numeric grade and details of all enrolled students living in state 'NY'.

**Outer Join**

1. Display the course number, description and pre-requisite along with section id. Those courses who do not have sections defined should also be displayed.

2. For the above query also include courses where no instructor is assigned.



**Self Join**

1. Many courses have pre-requisite. Display the course no and the description along with the prerequisite and the description. e.g. - 20 Intro to Java 10 Java Programming where 20 is the course no and 10 is the pre-requisite.

