# **Smart City**

**Use Case Scenario**

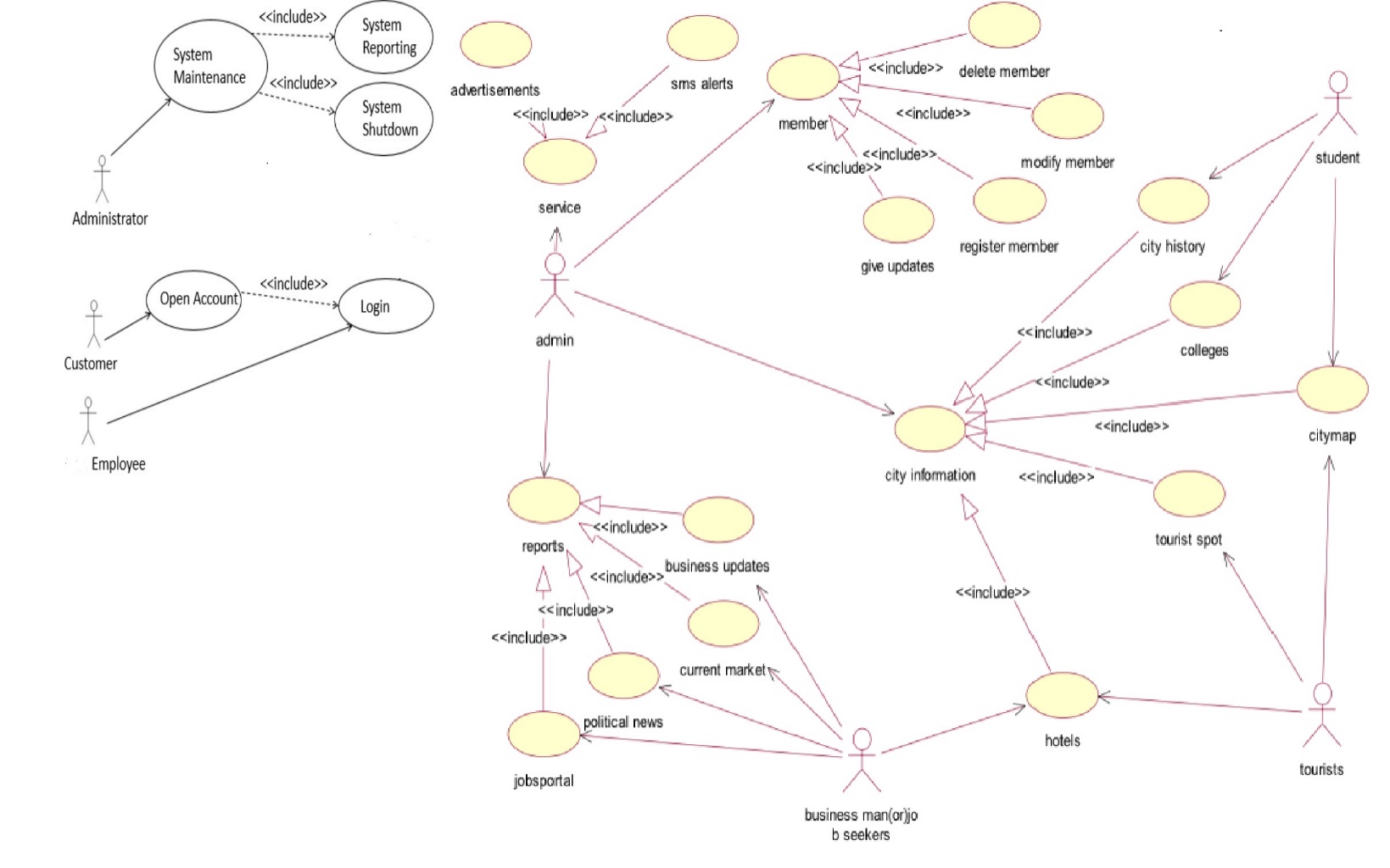
“NYC” has computer web-based Information management system. They offer online customer information about particular city account opening process. A customer can apply for account from anywhere in the world using web-based interface and submit all the required documents using online. The customer can create an account. After creating account customer can login by giving user name and password. For transaction customer must have login first,customer may give wrong PIN, then system will not authorize the customer for transaction. By transaction customer can perform certain things such as hotel booking for tourists, ticket booking, transport facility providing, business related information , marketing details, citynews ,shopping detail. It places to be visited, site maps route maps, business environment, job portal, information about organization that provide transport, hospitality and total history of the city. A employee can login into a customer account. Every account sytem has administrator whose job is to maintain the system. System maintenance must have two things one is syatem reporting and another systesm is shutdown.

It can be accessed by unlimited number of users. Each user will be assigned a different set of permissions for each module of the system. The user can have access to all the information in the site with limited services and provide extra services to registered users. Track all the transaction details of the customer. Confirmation of end user identity and will verify which users are authorized to receive Support. Maintain history of each customer and their related Maintain history of each customer and their related information.Every user must have their ID& PASSWORD for security purpose and AC.NO for transaction purpose. All the job seekers must have their Resume document to submit to administrator. Only registered members will be provided with communication between user, experts and general public through mails. Administrator is created in the system already. The administrator has to generate daily/weekly/Monthly reports, of the business and political news of the city. This site is best designed to be useful through internet to people of different places.

**Users of the System**

* Tourists,students,Businessmen
* Industrialists, Entrepreneur, Organizations academicians etc
* Jobseekers

**Use Case Diagrams:** A use case is a methodology used in system analysis in identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal.



**Use Case Diagram**

**Details description:**

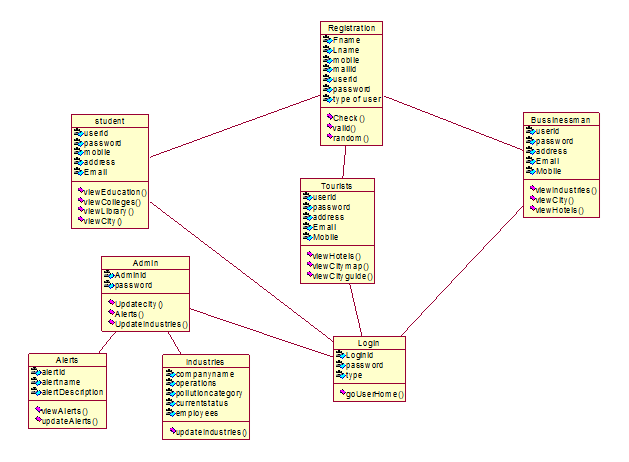
At first, we have to identify the actor of the existing scenario. Then we have to find out the actor, here the actor we have admin, tourists, students, businessmen. For denoting the actor, we use human sign. Generally, we use oval shape for activity, use **include** for must include activity, and **exclude** for may be activity.

**Name of the Use Case:** City Information

* Views the Required information pages.
* Click on register to get registration.
* Access required services through sign up into account.
* Views the reports generated by administrator
* Click on jobs to view jobs
* Click on business
* Login through User Id and password.

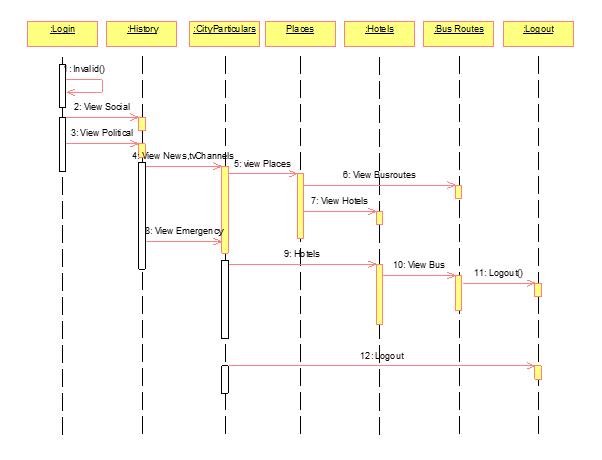
**Class Diagram:**

A class diagram is an illustration of the relationships and source code dependencies among classes in the unified modeling language. In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.



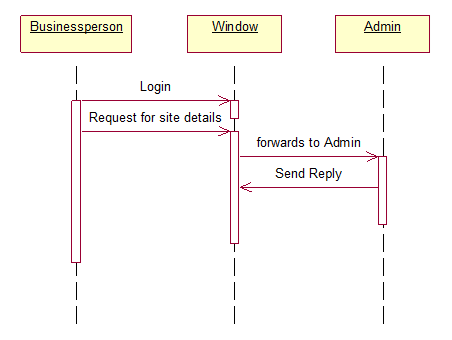
**Class Diagram of the System**

**Sequence Diagram:** Sequence diagram is a diagram that shows object interactions arranged in time sequence. In particular it shows objects participating in the interaction and the sequence of messages exchanged. For sequence diagram, first we need to identify the object of out=r system.



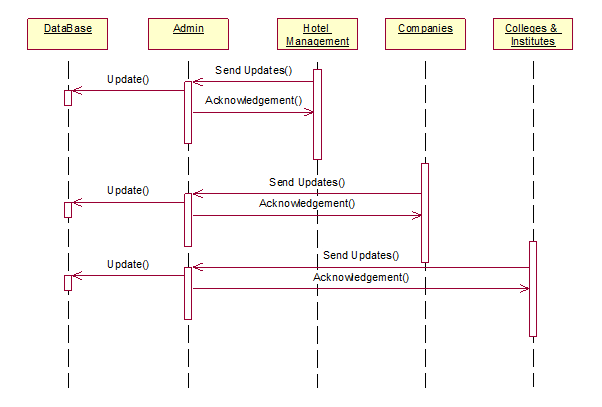
**Sequence Diagram of the User**

In above scenario, Login, History, Cityparticulars, Places, Hotels, Bus Routes, Logout are our system object.



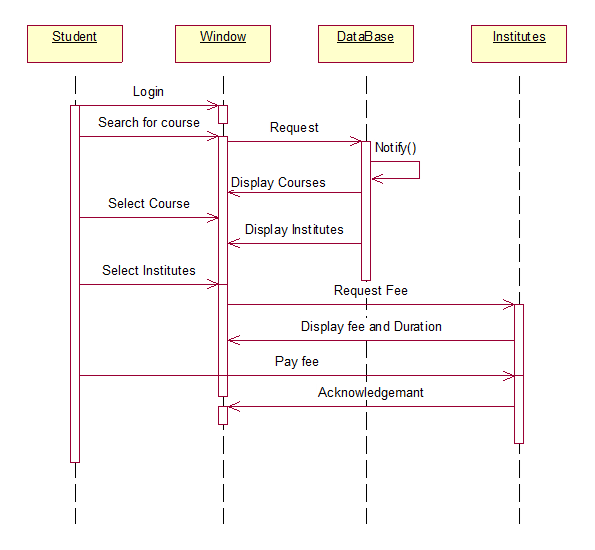
**Business Person Sequence Diagram**

In above scenario, Businessperson, Window and Admin are our system object.



**Admin Site Update Sequence Diagram**

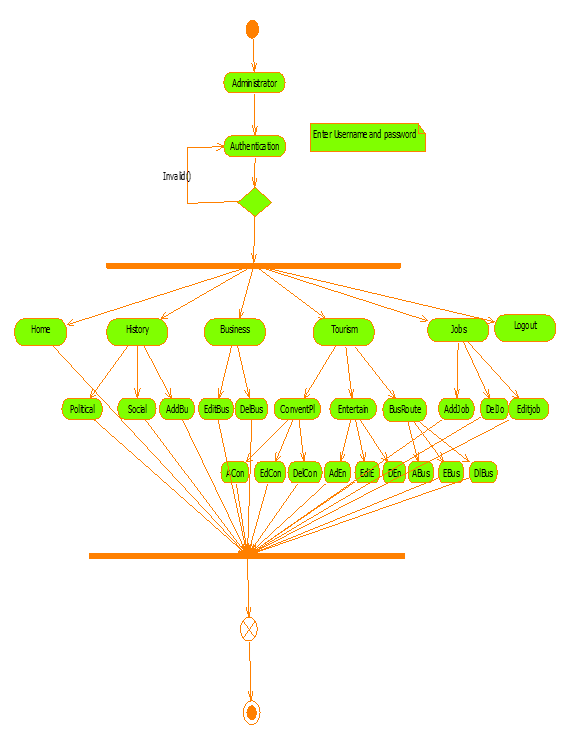
In above scenario, Database, Admin, Hotel Management, Companies and Colleges Institutes are our system object.



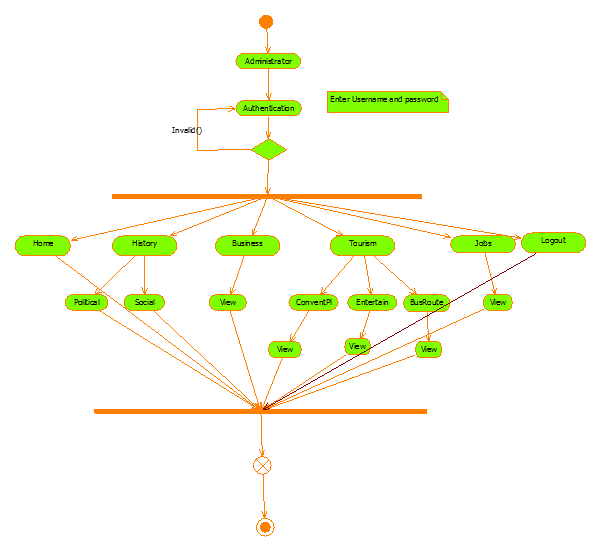
**Course Registration Sequence Diagram**

In above scenario, Student, Window, Database and Institutes are our system object.

**Activity Diagram:** Activity diagram describes various activities that takes places given in a particular process. The administrator activity diagram describes how the admin performs various activities in order to provide various services to the different types of users. The user activity diagram how the user will interact with the system and also various services that are accessible by him.



**Administrator activity diagram**

**User Activity Diagram**

**Details description:**

For activity diagram we use some symbol. We use diamond shape for denote decision making, black circle for start symbol where we start, oval shape for activity what we do, and black circle with white border for end where we end our activity.

For account checking of a customer an employee first enters the name of the account holder then give the account PIN number. Then the system will authenticate that the given account is correct or wrong. Then verify that information that it is valid or invalid if it is valid then it will check whether it has any city information. In case of wrong the system will promote reenter account then employee renter the account. If account is valid the system will go to check is there any service.

**System Specific Module**

There are five modules in our project.They are listed below with their description.

1. Administrator Module
2. Tourist Module
3. Student Module
4. Businessman Module
5. Jobseeker module

**1.Administrator Module:** The admin module is the major module as it is responsible for carrying out the major operations regarding site updates, business updates, job alerts etc., It maintains information regarding other four modules. The various software components in administrator module update alerts, update industries, update hotels, view resumes, update, site information. The details regarding complete history such as political and social is entered by admin and he want he can edit the details once entered. The details regarding business details such as top companies in the city and its information are entered and he want he can edit the details once entered. The details regarding job details such as job title, vacancies and companies' profile is entered by admin and he can edit the details once entered.

The details regarding emergency such as phone numbers of with respect to emergency is entered by admin and he can edit the details one he entered. The details regarding conventional places such as description an,location, address and image of the place entered by him and he can edit the details once entered

The details regarding newspapers, city channels such which papers are available in the city entered by admin and he can edit the details once he entered. The details regarding political leaders entered by admin and he can edit the details once he entered.

**2. Tourist Module:** Tourist module is maintaining the information regarding the city tourist spot, hotels in the city entertainment in this city etc..,the user after registration as a tourist is considered as authorized user. The various software components in the tourist module are view theaters, view hotels, view city map, view ATM locations, view hospitals, view city history, view travel agency, view bus routes.

**3.Jobseeker Module:** Jobseeker module is the commercial module as it need some paid services. The Jobseeker views the jobs portal and find a job availability. He then posts his resume to administrator. The various software components available in the jobseeker module are view jobs, view city information, view institutes, view city updates, post resume, view alerts, view city location.

**4.Businessman Module:** businessman module consists of information regarding various business in the city, industries in the city, with social and political influence of the city. The various software components in the businessman module are view city history, view markets, view alerts, view industries, view hotels, view Laboure, view hotels, view Laboure, view jobs.

**5. Student Module:** The student module maintains the various information regarding various institutes in the city. The various components in software are view library, view books, view institutes, view engineering colleges, view coaching centers, view journals.

The list of the software components involved in our project are explained below.

**a) View Map:** It displays the city map of the particular city by redirecting to google map and thus achieve to give the dynamic map.

**b) Insert Alerts:** Insert alerts into the database. This is used to give time to time alersts to the user.

**c)View Profile:** It display the information regarding different users who have registered.

**d) Bu. Search:** It provides the information regarding bus and timings if you give the source and destination.

**e) View Alerts:** It provides the alerts that are stored in database.

**f) Registration:** Provides registration facilities to different types of user.

**g) View Industries:** It provides information regarding their location and address with some more description attributes.

**h) View Hospitals:** It provides the information regarding hospitals in the city.

**i) Photo Gallery:** It provides different photos sightseeing places in the city.

**j) View City Guide:** It provide the city historical information with the static city map locating key places in the city.

**k) Add Emergency:** insert emergency reports into the data base. It provides emergency services that is needed by the user.

**l) Insert Industry:** it adds the information of industries in the database which provides the business person to see the number of industries in the city along with their description.

**m) Downloads:** It provides various information downloads that is needed to be downloaded by redirecting to best download site.

**n) Log Valid:** it validates the user id & password.

**o) View Resume:** The admin will view the resume posted by the user or jobseeker and forward to an organization.

**p) Flight Search:** it provides the information regarding flights the arrival and departure times.

**q) View Industries:** It provides information regarding their location and address with some more description attributes.

**r) View Hospitals:** It provides the information regarding hospitals in the city.

**s) Photo Gallery:** It provides different photos sightseeing places in the city.

**t) View City guide:** It provide the city historical information with the static city map locating key places in the city.

**u) Add Emergency:** insert emergency reports into the data base. It provides emergency services that is needed by the user.

**v) Insert Industry:** it adds the information of industries in the database which provides the business person to see the number of industries in the city along with their description.

**w)** **Downloads:** It provides various information downloads that is needed to be downloaded by redirecting to best download site.

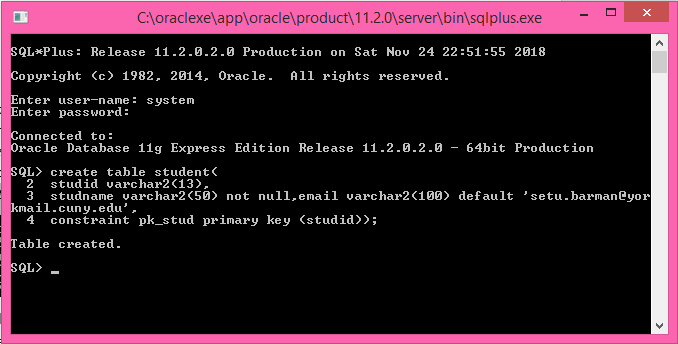
**x) View Map:** It displays the city map of the particular city by redirecting to google map and thus achieve to give the dynamic map.

**y) Insert Alerts:** Insert alerts into the database. This is used to give time to time alerts to the user.

**Data Base Students Table:**

1. Create a ***Student***table with necessary integrity constraint:

*create table student(*  
 *studid varchar2(13),*  
 *studname varchar2(50) not null,*  
 *email varchar2(100)* ***default*** [*’setu.barman@yorkmail.cuny.edu’*](mailto:‘shafikshaon@gmail.com’)*,*  
 *constraint pk\_stud primary key (studid));*



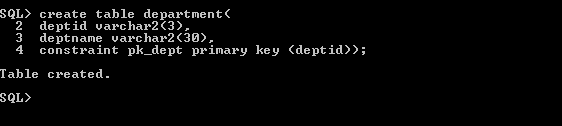
2. To display the physical structure of the table:

*desc student*

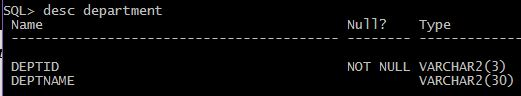


3. Create a ***department***table with necessary integrity constraint:

*create table department(*  
 *deptid varchar2(3),*  
 *deptname varchar2(30),*  
 *constraint pk\_dept primary key (deptid));*

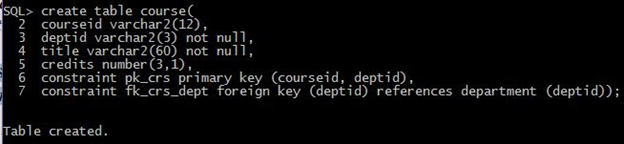


4. To display the physical structure of the department table:  
 *desc department*



5. Create a *course* table with ***courseid***as the primary key and ***deptid***as the foreign key from the

*department* table:  
 *create table course(*  
 *courseid varchar2(12),*  
 *deptid varchar2(3) not null,*  
 *title varchar2(60) not null,*  
 *credits number(3,1),*  
 *constraint pk\_crs primary key (courseid, deptid),*  
 *constraint fk\_crs\_dept foreign key (deptid) references department (deptid));*



6. Create a ***instructor***table with the necessary constraints:

*create table instructor(*

*instid varchar2(12),*

*deptid varchar2(3) not null,*

*instname varchar2(50) not null,*

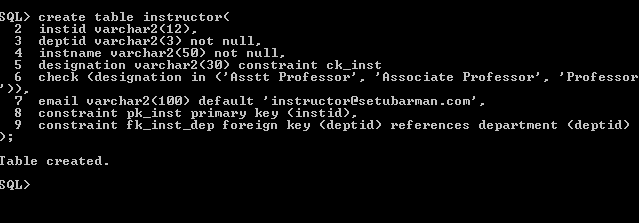
*designation varchar2(30) constraint ck\_inst*

*check (designation in ('Asstt Professor', 'Associate Professor', 'Professor')),*

*email varchar2(100) default '*[*instructor@setubarman.com*](mailto:instructor@shafikshaon.com)*',*

*constraint pk\_inst primary key (instid),*

*constraint fk\_inst\_dep foreign key (deptid) references department (deptid) );*



7. To display the structure of the table *instructor*:

*desc instructor*



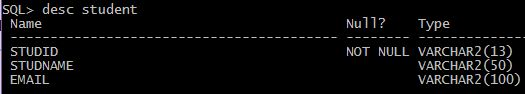
**Altering Table Structure (DDL)**  
 **8. Changing a column definition from** *not null* to *null*:

*alter table student modify (studname varchar2(50) null);*



9. To see the structure of the table *student*:

*desc student*



**Previous:**



10. Changing a column definition from *null* to *not null*:

*alter table student modify (studname varchar2(50) not null);*



11. To see the structure of the table *student*:

*desc student*



**Previous:**

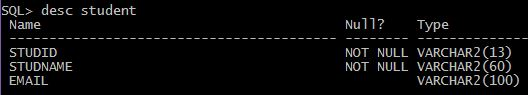


12. Increasing a column’s width:

*alter table student modify (studname varchar2(60) not null);*



13. To see the structure of the table *student*:  
 *desc student*



**Previous:**



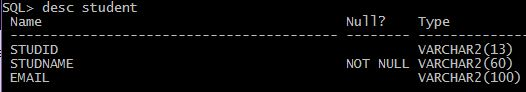
14. Deleting the Primary key from the table *student*:

*alter table student drop primary key;*



15. To see the structure of the table *department*:

*desc department*



**Previous:**



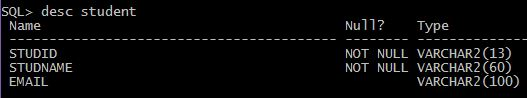
16. Adding Primary key for the table *student*:

*alter table student add constraint pk\_stud primary key (studtid);*



17. To see the structure of the table *student*:

*desc student*



Previous:

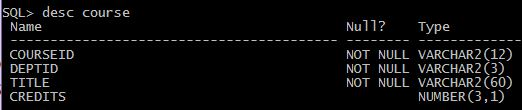


18. Deleting the foreign key from the table *course*:

*alter table course drop constraint fk\_crs\_dept;*



19. To see the structure of the table *course*:  
 *desc course*



20. Adding foreign key for the table *course*:  
   

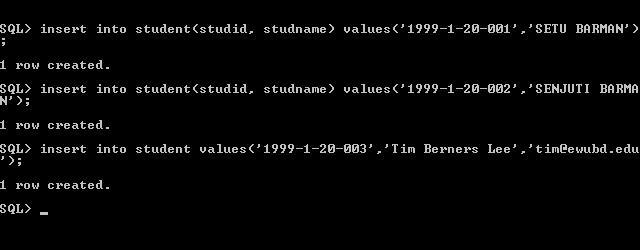

a*lter table course add constraint fk\_crs\_dept Foreign key (deptid) references department*  
 *(deptid);*

21. To see the structure of the table *course*:  
 *desc course;*



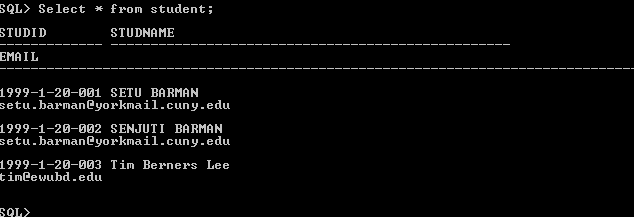
**Inserting some Rows of Information (DML)**  
 **22. Insert row of information in the** *student* table:

*insert into student(studid, studname) values(’1999-1-20-001’,’SETU BARMAN’);*  
 *insert into student(studid, studname) values(’1999-1-20-002’,’SENJUTI BARMAN’);*  
 *insert into student values(’1999-1-20-003’,’Tim Berners* [*Lee’,’tim@ewubd.edu’*](mailto:Lee’,’tim@ewubd.edu’)*);*



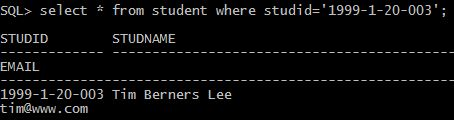
23. Display all the rows from the table *student*:

*Select \* from student;*



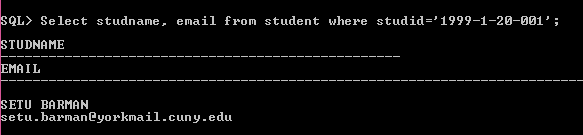
24. Display all the rows for the student *Tim BernersLee*:

*Select \* from student where studid=’1999-1-20-003’;*



25. Display the *name* and *email* of the student holding the ID#1999-1-20-001:

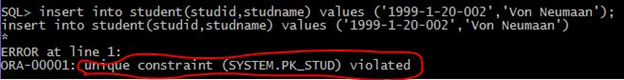
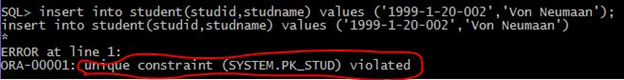
*Select studname, email from student where studid=’1999-1-20-001’;*



26. Insert the following row of information in the *student* table:

*Insert into student(studid, studname) values(‘1999-1-20-002’, ‘Von Neumaan’);*

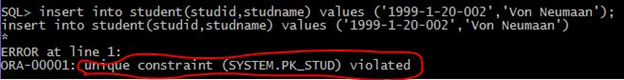
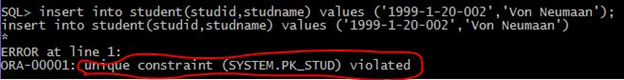
Please take note of the error message generated by the Oracle:  
 >>



27. Display the detail information of the student whose student ID is ‘1999-1-20-002’:

*Select stud, name, email from student where studid=’1999-1-20-002’;*

Please take note of the error message generated by the Oracle:  
 >>



Correct form of the SQL instruction:  
 >>

