Lab 1

1. Use interactive mode to create a database called BankAccount. Consider the BankAccount relational database with the following schema. Give an SQL DDL definition of this database. Identify referential-integrity constraints that should hold, and include them in the DDL definition.

   Customer (cust_id, fname, lname)

   Account (account_id, product_cd, cust_id, balance)

   Product (product_cd, name)

   Transaction (txn_id, txn_type_cd, account_id, amount, date)

   Mail (sentDateTime, srcuser, srchost, dstuser, dsthost, size)

   srcuser – sender 8 characters long

   srchost – the host for sender 20 characters long

   dstuser – recipient 8 characters long

   dsthost – the host for recipient 20 characters long

   a) Identify the primary keys, any alternate keys in all schema

   b) What is the degree of each relation?

2. Use script mode to create a database called stayHome. Consider the stayHome relational database with the following schema. Give an SQL DDL definition of this database. Identify referential-integrity constraints that should hold, and include them in the DDL definition.

   Branch (branchNo, street, city, state, zipCode, mgrStaffNo)

   Staff (staffNo, name, position, salary, branchNo)

   Video (catalogNo, title, category, dailyRental, price, directorNo)

   Director (directorNo, directorName)

   Actor (actorNo, actorName)

   Role (actorNo, catalogNo, character)

   Member (memberNo, fName, lName, address)

   Registration (branchNo, memberNo, staffNo, dateJoined)

   RentalAgreement (rentalNo, dateOut, dateReturn, memberNo, videoNo)

   VideoForRent (videoNo, available, catalogNo, branchNo)

3. To create a database called hotelBooking. Consider the hotelBooking relational database with the following schema. Give an SQL DDL definition of this database. Identify referential-integrity constraints that should hold, and include them in the DDL definition.

   Hotel (hotelNo, hotelName, City)

   Room (roomNo, hotelNo, type, price)

   Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

   Guest (guestNo, guestName, dateFrom, dateTo, roomNo)

   Where Hotel contains hotel details

   Room contains room details for each hotel

   Booking contains details of booking

   Guest contains guest details

4. Create a database called ClassSchedule. Consider the ClassSchedule relational database with the following schema. Give an SQL DDL definition of this database. Identify referential-integrity constraints that should hold, and include them in the DDL definition.

   Student (student_ID, student_name)

   PK: student_ID (integer data type)

   student_name (25 characters)

   Qualified (faculity_ID, course_id, date_qualified)

   PK: faculty_id, course_id

   Faculity (faculity_ID, faculity_name)

   PK: faculty_id (integer)

   faculty_name (25 characters)

   Section (section_NO, Semester, Course_ID)

   PK: section_no, semester, course_id, section_no (integer),

   semester (7 characters)

   Course (course_ID, course_name)

   PK: course_id (8 characters)

   course_name (15 characters)

Registration (student_ID, section_no, semester)

PK: student_id, section_no, semester

5. Create database called AdultLiteracy. The database tracks an adult literacy program.Tutors complete a certification class offered by the agency. Students complete an assessment interview that results in a report for the tutor and a recorded read score (between 0 to 10). When matched with a student, tutors meet with them from one to four hours per week. Some students work with the same tutor for years, some for less than a month, other students change tutors if their learning style does not match the tutor's tutoring style. Many tutors are retired and are only available to tutor for part of the year. Tutor status is recorded as Active, TempStop, or Dropped. Consider the AdultLiteracy relational database with the following schema. Give an SQL DDL definition of this database. Identify referential-integrity constraints that should hold, and include them in the DDL definition.

   Tutor (tutor_id, Create_date, status)

   PK: tutor_id

   Student (student_id, score)

   PK: student_id

   Match_History (match_id, tutor_id, student_id, start_date, end_date)

   PK: match_id