Case Study Scenario:

This case study is to develop a *Course Management System* (CMS) for ABC University. The following are the two use cases for which the database needs to be designed.

*Add Course*

*–* To add the course details into the course management system.

*Retrieve Course*

*–* Retrieve the courses stored in the system and display it.

The courses to be added will have the following attributes Course Code, Course Name, Number of participants, Course Description, Course Duration, Course start date and Course Type.

1. Let us develop the first table for our ABC university CMS system. Let us create the table for storing the course details. Create a table named “COURSE\_INFO” with following column name, data type and data size.

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Data Size |
| COURSE\_Code | int |  |
| COURSE | varchar | 20 |
| COURSE\_DESCRIPTION | varchar | 20 |
| COURSE\_START\_DATE | Date |  |

2. ABC university has come up with new requirements which requires a change in the COURSE\_INFO table we created.

**Requirement** # 1: The course code can be alpha numeric. Change the course code data type from numeric to varchar

**Requirement** # 2: Course description can have 200 characters. Change the course description data size from 20 to 200.

**Requirement** # 3: Course Info should capture three additional details

* Course\_Duration – numeric
* Number\_of\_Participants – numeric
* Course\_Type- Character(3)

**Requirement # 4:** Rename COURSE column as COURSE\_NAME.

3. **Requirement # 5:** ABC University architects have defined a standard to precede all the tables in the database with string “CMS\_”. Rename the table from COURSE\_INFO to CMS\_COURSE\_INFO.

**Requirement # 6:** ABC University decides to break the CMS\_COURSE\_INFO table into multiple small tables. Delete all the records in the table and drop the table from the database.

4. Let us insert the following four records into the “COURSE\_INFO” table which we had already created .

NOTE: If table is not there please recreate the table.

|  |  |  |  |
| --- | --- | --- | --- |
| COURSE\_CODE | COURSE\_NAME | COURSE\_DESCRIPTION | COURSE\_START\_DATE |
| 343 | Java Programming | Basics of Java | 12 /12/2012 |
| 167 | C Programming | Basics of C | 11/11/2012 |
| 347 | RDBMS | Basics Of RDBMS |  |
| 106 | Oracle | Oracle SQL | 11/03/2011 |

5. **Requirement # 7:** Change the Course name as “Oracle” to “MYSQL” and Description as “MySQL” for the course with code 106.

**Requirement # 8:** Remove the course RDBMS from the table.

|  |  |  |  |
| --- | --- | --- | --- |
| COURSE\_CODE | COURSE\_NAME | COURSE\_DESCRIPTION | COURSE\_START\_DATE |
| 343 | Java Programming | Basics of Java | 12 /12/2012 |
| 167 | C Programming | Basics of C | 11/11/2012 |
| 347 | RDBMS | Basics Of RDBMS |  |
| 106 | Oracle | Oracle SQL | 11/03/2011 |

6. Lets practice using select queries,

**Requirement # 9:** Insert few more records in the *course\_info* table. The green ones are newly inserted data. Check whether all rows can be inserted successfully.

|  |  |  |  |
| --- | --- | --- | --- |
| COURSE\_CODE | COURSE\_NAME | COURSE\_DESCRIPTION | COURSE\_START\_DATE |
| 343 | Java Programming | Basics of Java | 12 /12/2012 |
| 167 | C Programming | Basics of C | 11/11/2012 |
| 106 | MYSQL | My SQL | 11/03/2011 |
| 106 | Oracle | Oracle PL SQL | 11/03/2011 |
| 302 | Oracle | Oracle Architecture | 11/04/2011 |
| 231 | Core Java | Java IO | 11/06/2011 |

**Requirement # 10:** Retrieve courses whose name is “Core Java”

**Requirement # 11:** Retrieve courses whose name is “Core Java” and code is 343.

**Requirement # 12:** Retrieve the unique course names of the courses.