*Database Systems-I*

*CSC-371*

*PROJECT*

|  |  |
| --- | --- |
| **Name:** | Usama bin ghazan  Muhammad Ali Sheikh  Umar Ahmad |
| **Class:** | BSE-4b |
| **Registration Number:** | SP20-BSE-99  SP20-BSE-052  SP20-BSE-098 |
| **Instructor’s Name:** | Sir Basit Raza |

1. Contents

[Abstract 4](#_Toc91684728)

[3 Introduction: 4](#_Toc91684729)

[4 Problem Statement: 4](#_Toc91684730)

[5 Problem Solution/Objectives of the Proposed System: 4](#_Toc91684731)

[6 Vision Statement: 5](#_Toc91684732)

[7 Scope: 5](#_Toc91684733)

[8 Tools and Technologies: 5](#_Toc91684734)

[9 Project Stakeholders and Roles: 5](#_Toc91684735)

[10 Gantt Chart: 6](#_Toc91684736)

[11 Table Descriptions: 6](#_Toc91684737)

[12 Data Model: 8](#_Toc91684738)

[13 Functional Dependencies and Normalization: 9](#_Toc91684739)

[14 Data Definition Language (DDL) commands 13](#_Toc91684740)

[15 Tables: 24](#_Toc91684741)

[16 SQL QUERIES: 31](#_Toc91684742)

# Abstract

# Introduction:

One of the most important responsibilities of student management is manage and collect data of students. A student management system helps a school manage data, communications and scheduling. A school system generates and uses a large amount of data. This data must be communicated appropriately to students, faculty and parents. A student management system helps schools to store, manage and distribute this information. It also helps the department to register the student in course they have selected and perform many other desired functionalities.

# Problem Statement:

It is difficult to manage large amount of student data on a register a something physical. That’s why we need a software that can take care of all the problems regarding student data. This software should be able to store and manage student data efficiently.

# Problem Solution/Objectives of the Proposed System:

Our Student management system will store all the student data in the database and it will be very simple to retrieve this data without much effort.

**Objectives:**

The objective to be achieved from SMS is:

* To add student detail in the database.
* To delete student details from the database.
* To Search student details from the database.
* To update student details from the database.
* To display student details from the database

# Vision Statement:

Through this app we will be able to make work of every educational intuitional easy. They will no longer need to worry about managing long data of students and every detail will be stored at a single place.

# Scope:

In general student Management/Information is a system that stores data and enables functionality that organizes and maintains the student data within institutions databases. These systems may be an independent technology for the institution use only. The mini-world we have chosen is a student Management System in a school.

# Tools and Technologies:

Tools and Technologies for Proposed Project

|  |  |  |  |
| --- | --- | --- | --- |
| **Tools**  **And**  **Technologies** | **Tools** | **Version** | **Rationale** |
| MS Visual Studio | 2015 | IDE |
| MS SQL Server | 2015 | DBMS |
| Adobe Photoshop | CSC 6 | Design Work |
| **Technology** | **Version** | **Rationale** |
| Python | 3.9.7 | GUI Development |
| SQL Developer | 21.2.1.204.1703 | Back-end Development |

# Project Stakeholders and Roles:

Project Stakeholders for Proposed Project

|  |  |
| --- | --- |
| **Project Sponsor** | COMSATS University Islamabad, Islamabad Campus |
| **Stakeholder** | Usama Bin Ghazan, Umar Ahmad, Ali Sheikh   * Basit Raza |

# Gantt Chart:

***Graphical user interface

Description automatically generated***

# Table Descriptions:

**1. CLASSROOM:**

Keep information about class id(classroom\_id) and student grade(grade).

This table stores the Room no. assigned to a particular grade. Classroom\_id is primary key

**2. COURSE:**

Holds details about the course id(course\_id),course name(course\_name), course description(course\_description),teacher id(teacher\_id) , grade.

This table has a primary key for distinct courses in relation to the grade it is assigned to and the teacher that teaches it. Course name and description helps the end user to understand it

**3. EXAM:**

This table stores the information about the type of exams (Monthly, mid-term, annual etc) along with the specific date when it starts

**4. EXAM\_RESULT:**

Exam\_result table store the exam result of each student with percentage in each course.

**5. GRADE:**

This table contains only distinct Grades (1,2,3… upto matric or 8) which is primary key

**6. STUDENT:**

Contains attributes for student\_id(student\_id), first name(fname),last name(lname), date of birth of students(dob),(gender), phone numbers(phone), mobile numbers(mobile), joining date(date\_of\_join), address, grade

This table contains student’s personal information along with his grade (that is a foreign key from grade table). Student id is defined as primary key; means no 2 students can have same id

**7. TEACHER:**

This teacher table contains ids of teachers (teacher\_id), first name(fname),last name(lname), date of birth of students(dob), phone numbers(phone), mobile numbers(mobile).

This table stores teacher’s personal information. Teacher\_id is primary key here

**8. Attendance:**

Attendance table contains date , ids of student(student\_id), status.

This table stores everyday attendance status (Present or Absent) of each student along with the date

# Data Model:

**ER Diagram:**

Diagram

Description automatically generated

**Fig 1: ER diagram of Pharmaceutical-Retail Database System**

* **Relational Schema:**

Each relation is mapped using an appropriate foreign key keeping in mind the cardinality and participation constraints.

Diagram

Description automatically generated

**Fig 2: Relational Schema of Pharmaceutical-Retail Database System**

# Functional Dependencies and Normalization:

|  |  |  |  |
| --- | --- | --- | --- |
| **Relation** | **Functional Dependencies** | **Primary Key** | **Candidate Keys** |
| Class Room | classroom\_id → {grade} | classroom\_id | classroom\_id |
| STORES | course\_id → {course\_name,course\_description,teacher\_id,grade} | course \_id | course \_id |
| Exam | exam\_id → {exam\_type,stare\_date,} | exam\_id | exam \_id |
| Exam\_result | exam\_id → {student\_id,course\_id,percentage} | exam\_id | exam\_id |
| Grade | grade | grade | grade |
| Student | student\_id → { fname,lname,dob,gender,phone,  mobile,date\_of\_join,address,grade} | doc\_id | doc\_id, hos\_id |
| Teacher | teacher\_id → {{ fname,lname,dob,phone,mobile } | teacher\_id | teacher\_id |
| Attendance | student\_id → {date,status} | student\_id | student\_id |

**Normalization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **e\_id** | **e\_name** |  | **e\_address** | | | **e\_mobile** | |  |  |
| 101 | Sara |  |  | Lahore | | 03452312390 |  |  |  |
| 102 | Ali |  |  | Karachi | | 03002121212 |  |  |  |
|  |  | 04450012222 |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 103 | Romana |  | Rawalpindi | | | 03458881212 |  |  |  |
| 104 | Arslan |  | Peshawar | | | 03330000123 |  |  |  |
|  | 03323450987 |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **First Normal Form:** | | |  |  |  |  |  |  |  |
| • Each column should contain | | | |  |  |  |  |  |  |
|  | **e\_id** | **e\_name** |  | **e\_address** | **e\_mobile** |
| atomic values | | |  |  |  |
|  |  | 101 | Sara |  | Lahore | 03452312390 |
| • A column should contain | | |  |  |  |
|  |  | 102 | Ali |  | Karachi | 03002121212 |
| value of the same kind | | |  |  |  |
|  |  | 102 | Ali |  | Karachi | 04450012222 |
| • Each column should have a | | | |  |  |
|  | 103 | Romana |  | Rawalpindi | 03458881212 |
| unique name | | |  |  |  |
|  |  | 104 | Arslan |  | Peshawar | 03330000123 |
| • Order in which data is saved | | | |  |  |
|  | 104 | Arslan |  | Peshawar | 03323450987 |
|  |  |  |  |  |  |
| doesn’t matter | | |  |  |  |  |  |  |  |

All conditions for 1st normal form are satisfied except for the first one, as we can see that **e\_mobile** has multiple values. We can apply First Normal Form by separating the data in different rows.

|  |  |  |
| --- | --- | --- |
| **teacher\_id** | **subject** | **teacher\_age** |
| 401 | Maths | 34 |
| 402 | Physics | 32 |
| 403 | Biology | 28 |
| 404 | Physics | 30 |
| 405 | Chemistry | 37 |

**Second Normal Form:**

* It should be in 1st Normal Form
* It should not have any partial dependencies

We can see that the given table is in First Normal Form but **teacher\_age** is only dependent on **teacher\_id** and not the **subject** which leads to partial dependency. We can Apply Second NormalForm by separating the **Teacher** and **Subject** table as follows:

Teacher Table

|  |  |
| --- | --- |
| **teacher\_id** | **teacher\_age** |
| 401 | 34 |
| 402 | 32 |
| 403 | 28 |
| 404 | 30 |
| 405 | 37 |

**Subject Table**

|  |  |
| --- | --- |
| **teacher\_id** | **subject** |
| 401 | Maths |
| 402 | Physics |
| 403 | Biology |
| 404 | Physics |
| 405 | Chemistry |

# Data Definition Language (DDL) commands

-- phpMyAdmin SQL Dump

-- version 5.0.1

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1

-- Generation Time: Dec 22, 2020 at 06:32 PM

-- Server version: 10.4.11-MariaDB

-- PHP Version: 7.2.28

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `school\_1`

--

-- --------------------------------------------------------

--

-- Table structure for table `attendance`

--

CREATE TABLE `attendance` (

`date` date NOT NULL,

`student\_id` int(20) NOT NULL,

`status` enum('Present','Absent') NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `attendance`

--

INSERT INTO `attendance` (`date`, `student\_id`, `status`) VALUES

('2020-07-17', 1, 'Present'),

('2020-07-18', 1, 'Absent'),

('2020-07-17', 2, 'Present'),

('2020-07-18', 2, 'Present'),

('2020-07-17', 3, 'Absent'),

('2020-07-18', 3, 'Absent'),

('2020-07-17', 4, 'Absent'),

('2020-07-18', 4, 'Present'),

('2020-07-17', 5, 'Present'),

('2020-07-18', 5, 'Present'),

('2020-07-17', 6, 'Present'),

('2020-07-18', 6, 'Absent');

-- --------------------------------------------------------

--

-- Table structure for table `classroom`

--

CREATE TABLE `classroom` (

`classroom\_id` int(20) NOT NULL,

`grade` int(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `classroom`

--

INSERT INTO `classroom` (`classroom\_id`, `grade`) VALUES

(10, 1),

(11, 2),

(12, 3);

-- --------------------------------------------------------

--

-- Table structure for table `course`

--

CREATE TABLE `course` (

`course\_id` int(20) NOT NULL,

`course\_name` varchar(20) NOT NULL,

`course\_description` varchar(100) NOT NULL,

`teacher\_id` int(20) NOT NULL,

`grade` int(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `course`

--

INSERT INTO `course` (`course\_id`, `course\_name`, `course\_description`, `teacher\_id`, `grade`) VALUES

(110, 'English 1', 'Punjab Textbook Board', 103, 1),

(111, 'Urdu 1', 'Punjab Textbook Board', 100, 1),

(112, 'Maths 1', 'Federal Board', 104, 1),

(113, 'Science 1', 'Federal Board', 101, 1),

(114, 'Islamiyat 1', 'Punjab Textbook Board', 100, 1),

(220, 'English 2', 'Punjab Textbook Board', 103, 2),

(221, 'Urdu 2', 'Punjab Textbook Board', 100, 2),

(222, 'Maths 2', 'Federal Board', 104, 2),

(223, 'Science 2', 'Federal Board', 101, 2),

(224, 'Islamiyat 2', 'Punjab Textbook Board', 100, 2),

(330, 'English 3', 'Punjab Textbook Board', 103, 3),

(331, 'Urdu 3', 'Punjab Textbook Board', 100, 3),

(332, 'Maths 3', 'Federal Board', 104, 3),

(333, 'Science 3', 'Federal Board', 101, 3),

(334, 'Islamiyat', 'Punjab Textbook Board', 100, 3);

-- --------------------------------------------------------

--

-- Table structure for table `exam`

--

CREATE TABLE `exam` (

`exam\_id` int(20) NOT NULL,

`exam\_type` varchar(20) NOT NULL,

`start\_date` date NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `exam`

--

INSERT INTO `exam` (`exam\_id`, `exam\_type`, `start\_date`) VALUES

(98, 'Final-Term', '2020-12-01'),

(99, 'Mid-Term', '2020-06-15');

-- --------------------------------------------------------

--

-- Table structure for table `exam\_result`

--

CREATE TABLE `exam\_result` (

`exam\_id` int(20) NOT NULL,

`student\_id` int(20) NOT NULL,

`course\_id` int(20) NOT NULL,

`percentage` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `exam\_result`

--

INSERT INTO `exam\_result` (`exam\_id`, `student\_id`, `course\_id`, `percentage`) VALUES

(99, 1, 110, '90'),

(99, 1, 111, '89'),

(99, 1, 114, '95'),

(99, 1, 112, '70'),

(99, 1, 113, '80'),

(99, 2, 220, '98'),

(99, 2, 221, '82'),

(99, 2, 224, '87'),

(99, 2, 222, '90'),

(99, 2, 223, '70'),

(99, 3, 220, '20'),

(99, 3, 221, '10'),

(99, 3, 222, '50'),

(99, 3, 223, '25'),

(99, 3, 224, '78'),

(99, 4, 220, '56'),

(99, 4, 221, '27'),

(99, 4, 222, '24'),

(99, 4, 223, '19'),

(99, 4, 224, '59'),

(99, 5, 330, '79'),

(99, 5, 331, '55'),

(99, 5, 332, '87'),

(99, 5, 333, '89'),

(99, 5, 334, '50'),

(99, 6, 110, '88'),

(99, 6, 331, '78'),

(99, 6, 332, '59'),

(99, 6, 333, '40'),

(99, 6, 334, '57');

-- --------------------------------------------------------

--

-- Table structure for table `grade`

--

CREATE TABLE `grade` (

`grade` int(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `grade`

--

INSERT INTO `grade` (`grade`) VALUES

(1),

(2),

(3);

-- --------------------------------------------------------

--

-- Table structure for table `student`

--

CREATE TABLE `student` (

`student\_id` int(20) NOT NULL,

`fname` varchar(20) NOT NULL,

`lname` varchar(20) NOT NULL,

`dob` date NOT NULL,

`Gender` enum('Male','Female') NOT NULL,

`phone` int(15) NOT NULL,

`mobile` int(11) NOT NULL,

`date\_of\_join` date NOT NULL,

`address` varchar(50) NOT NULL,

`grade` int(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `student`

--

INSERT INTO `student` (`student\_id`, `fname`, `lname`, `dob`, `Gender`, `phone`, `mobile`, `date\_of\_join`, `address`, `grade`) VALUES

(1, 'Ahmad', 'Ali', '2010-03-05', 'Male', 51001001, 3001001, '2014-01-01', 'h#001 st#001 RWP', 1),

(2, 'Sadia ', 'Aslam', '2010-07-01', 'Female', 51002002, 3002002, '2014-01-01', 'h#002 st#002 RWP', 2),

(3, 'Ali', 'Murtza', '2009-03-01', 'Male', 51003003, 3003003, '2013-01-01', 'h#003 st#003 RWP', 2),

(4, 'Waqas', 'Raza', '2009-03-09', 'Male', 51004004, 3004004, '2013-01-01', 'h#004 st#004 RWP', 2),

(5, 'Naila', 'Amir', '2008-02-03', 'Female', 51005005, 3005005, '2012-01-01', 'h#005 st#005 RWP', 3),

(6, 'Ahsan', 'Khan', '2008-03-02', 'Male', 51006006, 3006006, '2012-01-01', 'h#006 st#006 RWP', 3);

-- --------------------------------------------------------

--

-- Table structure for table `teacher`

--

CREATE TABLE `teacher` (

`teacher\_id` int(20) NOT NULL,

`fname` varchar(20) NOT NULL,

`lname` varchar(20) NOT NULL,

`dob` date NOT NULL,

`phone` int(20) NOT NULL,

`mobile` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `teacher`

--

INSERT INTO `teacher` (`teacher\_id`, `fname`, `lname`, `dob`, `phone`, `mobile`) VALUES

(100, 'Asif', 'Khan', '1980-03-06', 51100100, 3100100),

(101, 'Sadia ', 'Sultan', '1990-02-09', 51101101, 3101101),

(103, 'Haroon', 'Rasheed', '1989-04-02', 51103103, 3103103),

(104, 'Tabinda', 'Naz', '1980-04-02', 51104104, 3104104);

--

-- Indexes for dumped tables

--

--

-- Indexes for table `attendance`

--

ALTER TABLE `attendance`

ADD KEY `student\_id` (`student\_id`);

--

-- Indexes for table `classroom`

--

ALTER TABLE `classroom`

ADD UNIQUE KEY `classroom\_id` (`classroom\_id`),

ADD KEY `grade` (`grade`);

--

-- Indexes for table `course`

--

ALTER TABLE `course`

ADD PRIMARY KEY (`course\_id`),

ADD KEY `teacher\_id` (`teacher\_id`),

ADD KEY `grade` (`grade`);

--

-- Indexes for table `exam`

--

ALTER TABLE `exam`

ADD PRIMARY KEY (`exam\_id`),

ADD UNIQUE KEY `exam\_id` (`exam\_id`);

--

-- Indexes for table `exam\_result`

--

ALTER TABLE `exam\_result`

ADD KEY `student\_id` (`student\_id`),

ADD KEY `course\_id` (`course\_id`),

ADD KEY `exam\_id` (`exam\_id`);

--

-- Indexes for table `grade`

--

ALTER TABLE `grade`

ADD PRIMARY KEY (`grade`);

--

-- Indexes for table `student`

--

ALTER TABLE `student`

ADD PRIMARY KEY (`student\_id`),

ADD UNIQUE KEY `student\_id` (`student\_id`),

ADD KEY `grade` (`grade`);

--

-- Indexes for table `teacher`

--

ALTER TABLE `teacher`

ADD PRIMARY KEY (`teacher\_id`),

ADD UNIQUE KEY `teacher\_id` (`teacher\_id`);

--

-- Constraints for dumped tables

--

--

-- Constraints for table `attendance`

--

ALTER TABLE `attendance`

ADD CONSTRAINT `attendance\_ibfk\_1` FOREIGN KEY (`student\_id`) REFERENCES `student` (`student\_id`) ON UPDATE CASCADE;

--

-- Constraints for table `classroom`

--

ALTER TABLE `classroom`

ADD CONSTRAINT `classroom\_ibfk\_1` FOREIGN KEY (`grade`) REFERENCES `grade` (`grade`);

--

-- Constraints for table `course`

--

ALTER TABLE `course`

ADD CONSTRAINT `course\_ibfk\_1` FOREIGN KEY (`teacher\_id`) REFERENCES `teacher` (`teacher\_id`),

ADD CONSTRAINT `course\_ibfk\_2` FOREIGN KEY (`grade`) REFERENCES `grade` (`grade`);

--

-- Constraints for table `exam\_result`

--

ALTER TABLE `exam\_result`

ADD CONSTRAINT `exam\_result\_ibfk\_1` FOREIGN KEY (`exam\_id`) REFERENCES `exam` (`exam\_id`),

ADD CONSTRAINT `exam\_result\_ibfk\_2` FOREIGN KEY (`student\_id`) REFERENCES `student` (`student\_id`),

ADD CONSTRAINT `exam\_result\_ibfk\_3` FOREIGN KEY (`course\_id`) REFERENCES `course` (`course\_id`);

--

-- Constraints for table `student`

--

ALTER TABLE `student`

ADD CONSTRAINT `student\_ibfk\_1` FOREIGN KEY (`grade`) REFERENCES `grade` (`grade`);

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

# Tables:

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

# SQL QUERIES:

Graphical user interface, text, application

Description automatically generated

**Range condition using the between operator with where clause:**

A screenshot of a computer

Description automatically generated

**Using Like operator to find name**

**A screenshot of a computer

Description automatically generated**

**Using Aggregate function for minimum, maximum, average**

**A screenshot of a computer

Description automatically generated**

**Joins**

Graphical user interface, text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated