

EduCat

Project on DBMS-Fall 2015

Team Name: Inoent

Team Members

Jiaqi Yu

Prakash Barri

Karthik Mohan

Pawan Kumar Murty Chittur

Table of Contents

Table of Contents	i
1. Introduction.....	1
2. Conceptual Database Design – Requirement Analysis.....	2
2.1. Courses	2
2.2. Course Categories	2
2.3. Course Materials	3
2.4. Course Trends	3
2.5. Course Impact	3
2.6. Rating	4
2.7. Browsing	4
2.8. Searching	5
2.9. Bidding	6
3. Conceptual design -Entity-Relationship Model	7
3.1. EduCat- Architecture	7
3.2. Core division of EduCat.....	7
3.3. IT division of EduCat	10
4. Schema Refinement and Normalization	15

4.1.Schema refinement.....	15
4.2.Normalization.....	16
5. SQL STATEMENTS	26
<i>4.1. SQL TABLE Creation and Loading data: - Core Model</i>	<i>26</i>
6. Technology Survey.....	29
<i>5.1. DESIGN APPROACH.....</i>	<i>29</i>
<i>5.2. TECHNOLOGY</i>	<i>29</i>
7. POPULATE DATA FOR DATABASE	31
<i>6.1. CORE MODEL</i>	<i>31</i>
<i>6.2. HR MODEL</i>	<i>34</i>
<i>6.3. PR MODEL</i>	<i>36</i>

1. Introduction

In this era of Internet dominating the way we perceive and receive information, it is an understatement if we point out that the entire world has transformed into a “global village”. People at any part of the world have the privilege to access the same information as the rest of the world. The evolution of Internet have brought us to a point wherein we happily upload to and download information from a central repository. We term it as “Cloud”. In our project, we are focusing on using this feature to deliver the “Most fundamental right of Humanity” to humanity itself. It is the “Right to Education”.

Our project derives its motivation from websites like Coursera, eDx and How Stuff Works .These websites act as a platform wherein learning materials are available online with same quality as it would be on a real-time classroom. In addition to that, anybody who has access to internet in any part of the world can access these websites provided they have the interest to learn. In this project, we have derived ideas from many such websites and are setting foot on a new project called “EduCat”. Our primary goal is to develop the infrastructure for Database Management Systems (DBMS) with this idea as the inspiration behind that.

2. Conceptual Database Design – Requirement Analysis

2.1. Courses

Courses in “EduCat” are offered by Top notch Industrial sectors and the World’s best universities who are already involved in extra-ordinary research in that particular domain of study. So each course will be designed by the Course Providers in such a way that it offers tremendous insights into that subject based on their experiences and learnings. Basic courses are offered for free of cost and advanced courses are offered for a reasonable fee. Students performing well in the mid terms of the course also have a possibility to get discounts to pay the portion of the remaining fee. Each course carries a unique course id (cid) which will be used starting from course registration until certification of the course. Course Schedule, duration of the course, starting and ending dates of the course will be decided and updated on the website by the instructor of the course. Creating a login id in the website is mandatory and it will be used to uniquely identify the registered user. Course list will be constantly updated and the currently active set of courses will be uniquely represented for the ease of users

2.2. Course Categories

EduCat offers wide variety of courses spanning Architecture, Arts & Culture, Biology and life sciences, Business and Management, Chemistry, Engineering, Economics, Law, Philosophy, Social Sciences etc. These courses form the main categories and are accessed in the form of a

tree with the complete set of courses as the root node. Course categories are expanded in such a way that user will be able to reach his desired category within 2-3 clicks.

2.3. Course Materials

Every course needs to be assisted with resources such as text books, reference books, online labs, online video tutorials and then infrastructure for assignment and project management. A database which will track with the number of videos covering one lecture and its duration and the progress of course work also needs to be done

2.4. Course Trends

A new feature has been added in “EduCat” termed “Course Trends”. In this, we will maintain a database of most wanted courses and most searched for keywords and the high-in-demand certifications. Feedback of this will be taken into account and the information will be passed on to the instructor to start the course which is in high demand.

2.5. Course Impact

Another new feature added in our project is the Course Impact metric. This keeps the history of highly participated courses in the past and keeps the statistics of number of certifications granted, number of employers who offered jobs based on that and testimonials of past registered users. This will help current users to gauge the value of the courses

2.6. Rating

Rating in our system is a relationship not entity. The rating happens between user and course. The rate can be described as numbers of stars. And the numbers of stars represent the user's satisfaction to this course. Meanwhile, the user can also write his description of feeling.

For example, a student thinks this course is hard to understand and the professor performs badly, he is very angry to this course. So he gives this course one star, and he writes some words to describe his feeling. If a student thinks the course is just okay, maybe he will give it three stars and if someone likes it very much, he can give the course full of five stars in our system.

In our system, different kinds of user can both rate the course from his perspective.

2.7. Browsing

Browsing function in our system is neither entity nor relationship. We achieve it by traverse the database. In our system, what user can browse are courses, course records, institutions, professors, schedules and rates in figure 2.

For example, if a user is browsing the system. He can see what course do we have, the course's supplier and professor, the course's schedule and how people rate this course or some other information.

For registered users: Registered user has the same authority with user.

For institution: Besides what user can do, institution can also browse details of the courses taught by its professors, but it has no authority to see the other professors' courses' record details.

For example, it can browse the course records of what its professors teach, to see the situation of course registration.

For professor: Besides what user can do, professor can also browse details of this professor's courses. He has no authority to see the other courses' record details.

For example, it can browse the course records of what he teaches, to see the situation of course registration.

2.8. Searching

Searching function in our system is neither entity nor relationship. We achieve it by search the database. In our system, user can search course, course records, institutions, professors, schedules and rates in figure 2, and user can search by many methods.

For example, if a user is searching the system. He can search the course by specific name, or he can just search by the category, specific professor, institution, etc. He can also search the rates about a specific kind of courses, to find the better courses.

For registered users: Registered user has the same authority with user.

For institution: Besides what user can do, institution can also search details of the courses taught by its professors, but it has no authority to search the other professors' courses' record details.

For professor: Besides what user can do, professor can also search details of this professor's courses. He has no authority to search the other courses' record details.

2.9. Bidding

Bidding in our system is neither entity nor relationship. The bidding happens between user and course, we can finish it in the register relation.

Normally, the professor's energy is limited, as a teacher he needs to grade students' homework and answer students' question. So sometimes the course can be bid as a selling item. To this kind of courses, we need the bidding function.

We achieve it in the register relation. Everyone who wants to attend this course will give his bid, and the system records it. After this, if the course has n seats, we choose the higher n registered users to register.

3. Conceptual design -Entity-Relationship Model

3.1. EduCat- Architecture

For the ease of understanding, we have split the database design into 4 categories each modelled with supporting ER diagrams and explanations. They are as follows

1. Core Model or Sales Model (CM)
2. Human Resources Model (HRM)
3. Information Technology Model (ITM)
4. Public Relations Model (PRM)

3.2. Core division of EduCat

Core division of EduCat comprises of entities and relationships related to Courses, different categories, course resources and added new features like Course Trending and Course Impact.

This ER diagram in Figure 1 conveys the following inferences. The main entities are Courses, Impact and Categories. Weak entities are Trends and Resources. They are all connected by following relationships. Courses are split into categories. Courses are measured by the impact index. Courses are famed by the trends and finally courses are assisted by resources.(Note that attributes are not represented in Figure 1 and 2 in order to have clarity in diagram.

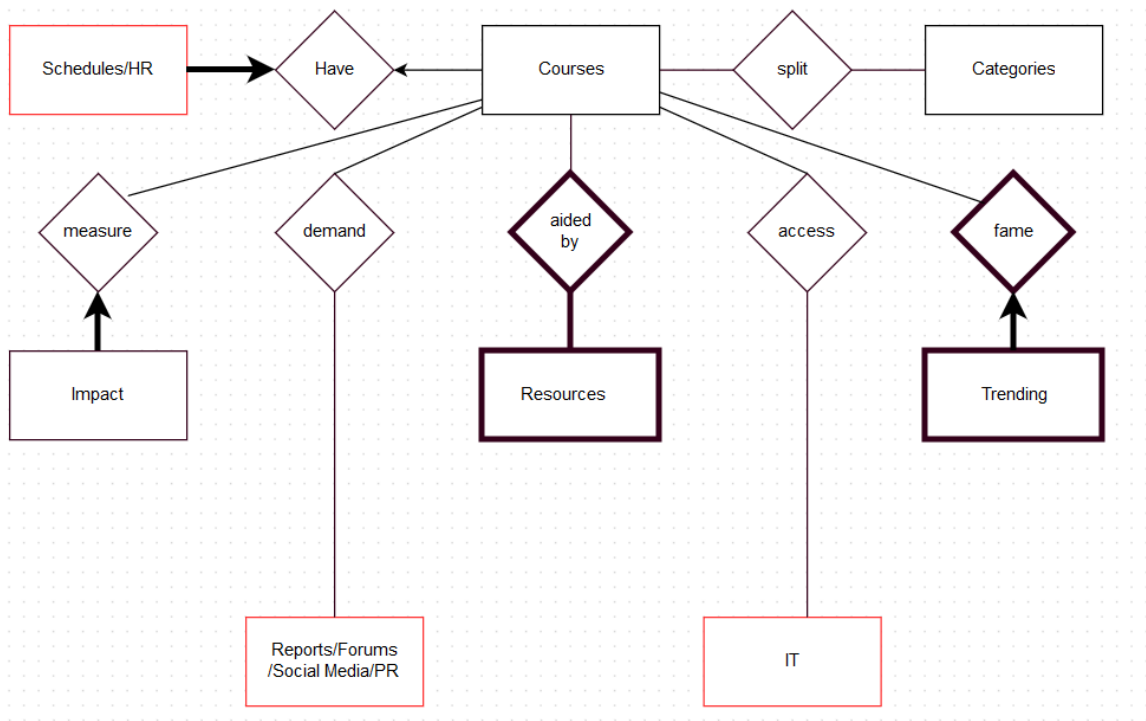


Figure 1 . EduCat – Core Model ER diagram (Colored entities belong to other model)

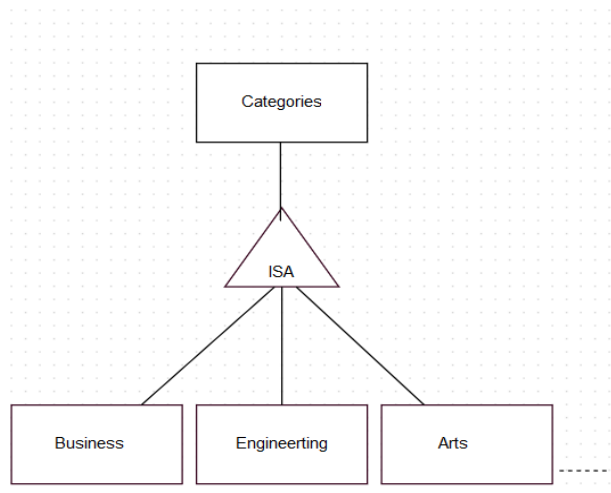


Figure 2 EduCat – Core Categories Hierarchy

Figure 2 explains about the different categories and their sub categories. They form an ISA relationship with the parent node. They form a tree structure so that it can be used during browsing of courses

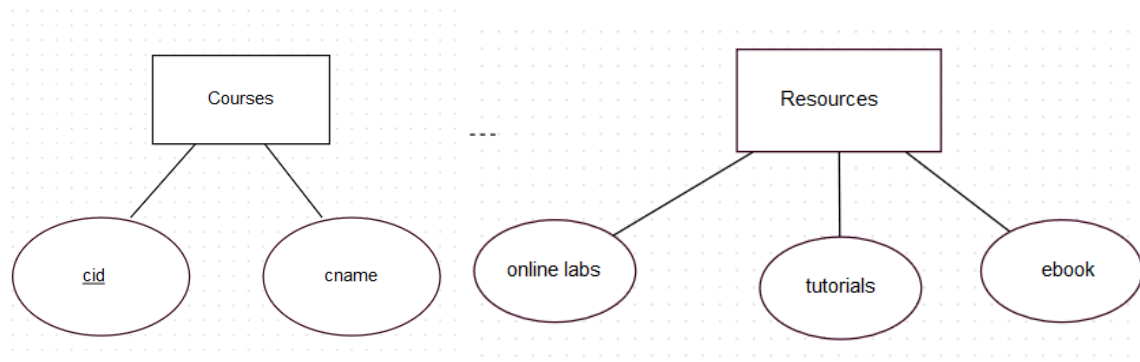


Figure 3: Entity-attribute set for Courses and resources

Figure 3 represents a sample entity- attribute set for courses and resources separately. The relationship between them is already represented in Figure 1.

3.3. IT division of EduCat

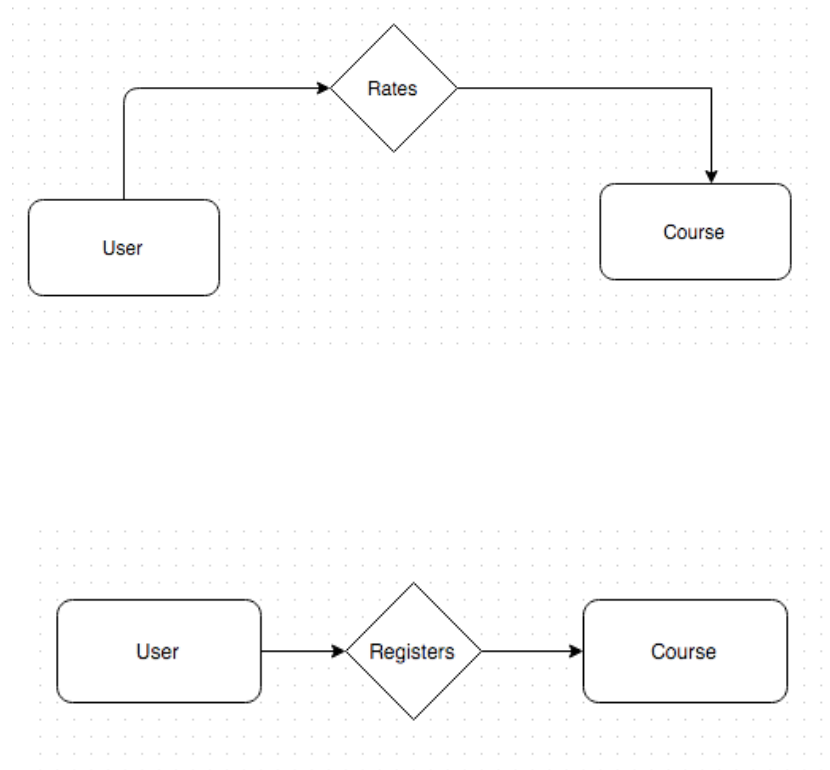


Figure 4. ER diagrams for IT model of EduCat

3.4 EDUCAT Social

EDUCAT social is a functionality offered by EDUCAT that allows users to share details about registered courses and course offerings in social networking sites such as face book, twitter and google plus.

Below E-R Diagram describes the entities in the Social Media module, Each registered user who completes the course can share his/her course certificate in professional websites such as Linked in, Indeed and so on and he can share course offering details and course details in social media sites such as Facebook, Twitter and so on.

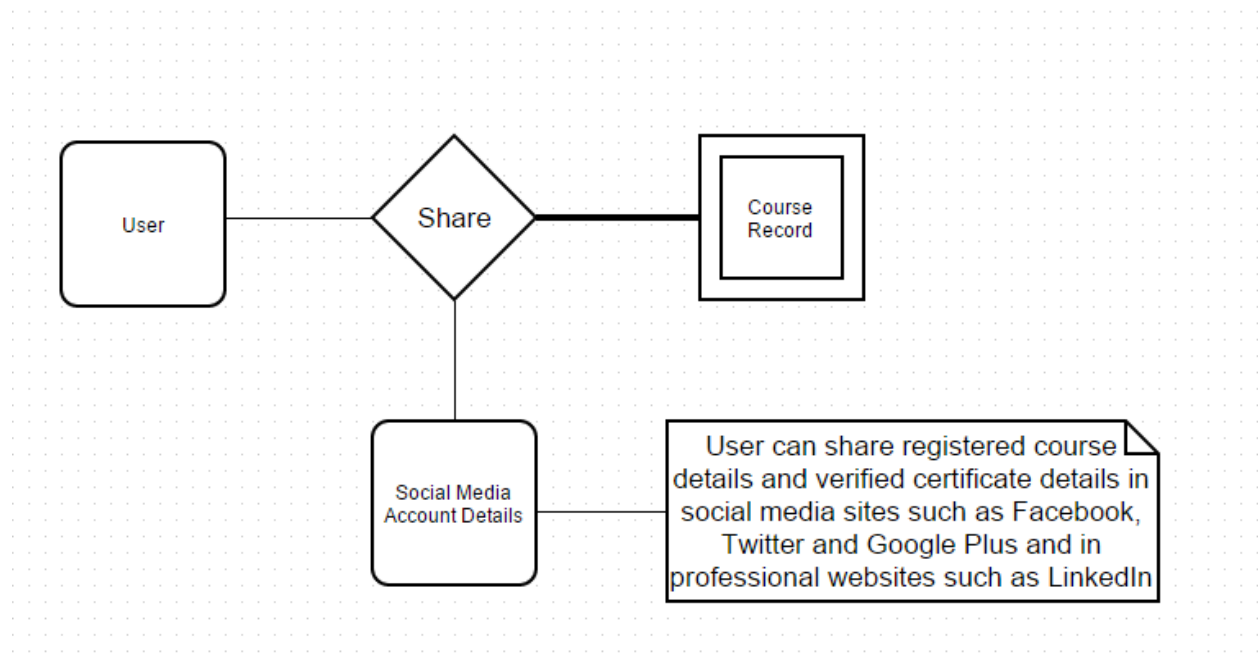


Figure 5: E-R Diagram for EDUCAT Social

3.5 EDUCAT Forum

EDUCAT forum is a functionality offered by EDUCAT that allows registered users to collaborate with other registered users. In a broader perspective each registered user can

post his/her questions in the forum, fellow users and professors can reply back to the questions posted by the users, they can share their valuable insights and so on. Each forum is specific to a course, each forum will have sub forums which categorized based on the labels such as exercise forums, assignment forums and so on.

Below E-R Diagram describes the entities in the Course Forum, Each offered course has its own forum through which students and professors can collaborate. Each registered user posts his/her questions about the course in the respective forums, his/her fellow users and professors reply back to the users questions.

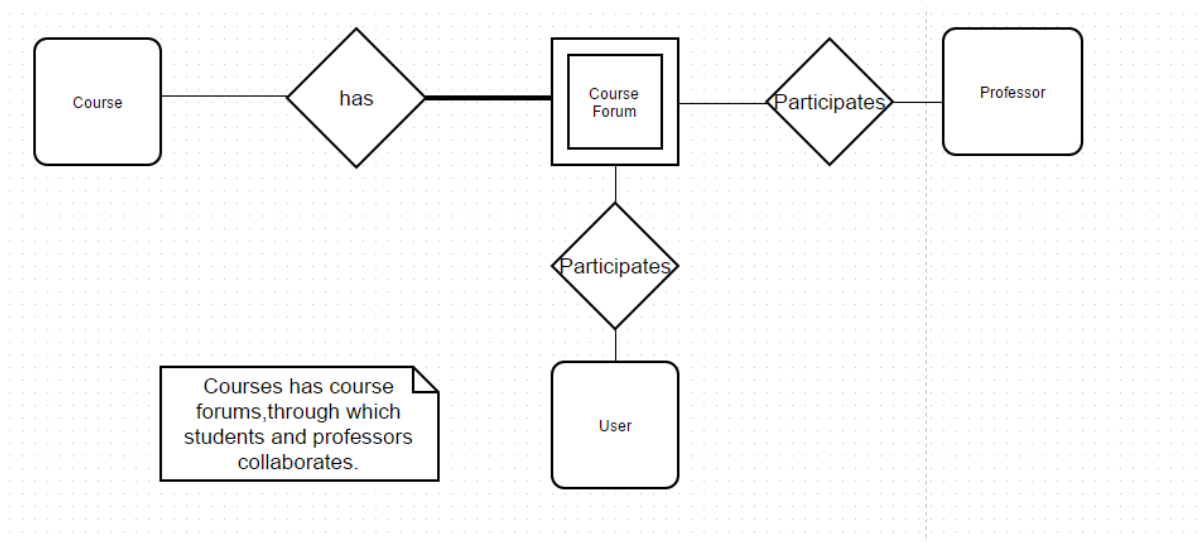


Figure 6: E-R Diagram for EDUCAT forum

3.6 EDUCAT Tracker

EDUCAT tracker is a functionality that tracks each user's progress, course progress and professor's course schedule details, based on the data the tracker creates valuable insights such as number of users registered for a course, how many of them are dropped in between, feedback about this particular course and so on.

This E-R diagram describes the entities in the course tracker system, course tracker system captures information about the course, course and user progress and other valuable insights such as grades, drop outs and how many attempts each user took to clear the online tests and so on.

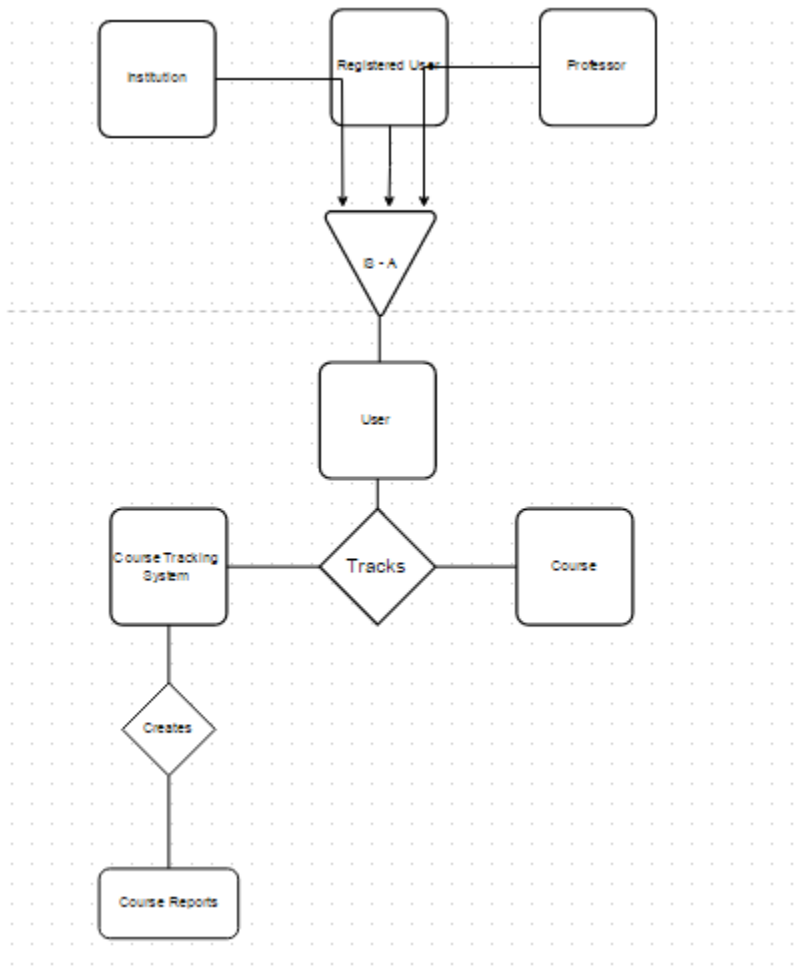


Figure 7: Macro level E-R diagram for EDUCAT tracker

4. Schema Refinement and Normalization

4.1.Schema refinement

With ER diagram as the reference, Schema for the Core model is derived here

Initial Schema:

Course(cid:integer,cname:string,cat_id:integer,pid:integer,pname:integer,pinst_name:string,sid:integer,sname:string)

Resources(cid:integer,rack_id:integer,rtype:string,rname:string,keyword:string)

Assignment(cid:integer,astart:date,aend:date,ano:integer)

Impact (cname:string,keyword:string,II:real)

Trend(cname:string,TI:real)

Rating(regid:integer, [cid:integer](#), score:integer, content:string)

Bidding(regid:integer, [cid:integer](#), ctime:date, price:real)

Browsing(regid:integer, keyword:string)

Schema definition: (in order of assignment)

Course →

(course_id,course_name,category_id,professor_id,professor_name,Institution_name,student_id,student_name)

Resources →

(course_id,rack_id,rack_type,rack_name,keyword)

Assignment →

(course_id,assignment_start_date,assignment_end_date,assignment_no)

Impact →

(Course_Name,keyword,Impact_Index)

Trend →

(course_name,Trend_Index)

Rating →

(reg_user_id,course_id,score,content)

Bidding →

(reg_user_id,course_id,chattime,price)

Browsing →

(reg_user_id,keyword)

4.2.Normalization

1NF Test on Schema:

Course(cid:integer,cname:string,cat_id0:integer,cat_id1:integer,cat_id2:integer ,pid:integer,pname:integer,pinst_name:string,sid:integer,sname:string)

Category0(cat_id0:integer,cat_name0:string)

Category1(cat_id1:integer,cat_name1:string)

Category2(cat_id2:integer,cat_name2:string)

Resources(cid:integer,rack_id:integer,rtype:string,rname:string,keyword:string)

Assignment(cid:integer,astart:date,aend:date,ano:integer)

Impact (cname:string,keyword:string,II:real)

Trend(cname:string,TI:real)

Rating(regid:integer, [cid:integer](#), score:integer, content:string)

Bidding(regid:integer, [cid:integer](#), ctime:date, price:real)

Browsing(regid:integer, keyword:string)

1NF Summary of Changes:

In the above set of schema, cat_id is found to be multivalued. Our database is planned in such a way that a normal browsing takes only 3 clicks before which we reach the desired section. So we will be in requirement of 3 categories which will be referred hierarchically.

Category0(Main category)

→ Category2(subcategory1)

-->Category 3(subcategory2)

Hence separate entities are created for each categories thereby **passing the 1NF test**

2NF Test on Schemas:

Course(cid:integer,cname:string,cat_id0:integer,cat_id1:integer,cat_id2:integer)

Professor(pid:integer,pname:integer,pinst_name:string,cid:integer)

Student(sid:integer,sname:string,cid:integer)

Category0(cat_id0:integer,cat_name0:string)

Category1(cat_id1:integer,cat_name1:string)

Category2(cat_id2:integer,cat_name2:string)

Resources(cid:integer,rack_id:integer,rtype:string,rname:string,keyword:string)
Assignment(cid:integer,astart:date,aend:date,ano:integer)
Impact (cname:string,keyword:string,II:real)
Trend(cname:string,TI:real)

Rating(regid:integer, [cid:integer](#), score:integer, content:string)
Bidding(regid:integer, [cid:integer](#), ctime:date, price:real)
Browsing(regid:integer, keyword:string)

2NF Summary of Changes:

In the above set of Schema, 2NF checks resulted in splitting the 'Course' schema into 3 parts – Course, Professor, Student because in 2NF check all the non-prime attributes must be determined by each one of the candidate keys. In above scheme (cid,pid,sid) forms the candidate key and here 'cid' does not determine 'sname'. Hence 2NF check fails and we have split the schema into 3 schemas

3NF Test on Schemas:

Course(cid:integer,cname:string)
Category(cid:integer,cat_id0:integer,cat_id1:integer,cat_id2:integer)
Professor(pid:integer,pname:integer,pinst_name:string,cid:integer)
Student(sid:integer,sname:string,cid:integer)
Category0(cat_id0:integer,cat_name0:string)
Category1(cat_id1:integer,cat_name1:string)
Category2(cat_id2:integer,cat_name2:string)
Resources(cid:integer,rack_id:integer,rtype:string,rname:string,keyword:string)
Assignment(cid:integer,astart:date,aend:date,ano:integer)
Impact (cname:string,keyword:string,II:real)
Trend(cname:string,TI:real)

Rating(regid:integer, [cid:integer](#), score:integer, content:string)
Bidding(regid:integer, [cid:integer](#), ctime:date, price:real)
Browsing(regid:integer, keyword:string)

3NF Summary of Changes:

In the above set of schemas, Course schema is split into "Course" and "Category" schemas because there existed a functional dependency between "cname" and "cat_id0","cat_id1" and "cat_id2" because the course name irrespective of whomsoever offers it will belong to the same category and subcategories. So here it fails 3NF check and so it is split into two schemas.

Note: All the schemas here have been designed in such a way that there is no overlapping candidate key. Henceforth all these schemas are also in BCNF form

So the final schema which is of BCNF form is as below

Course (cid:string,cname:string)

Category (cid:string,cat_id0:string,cat_id1:string,cat_id2:string)

Professor (pid:string,cid:string)

Student (sid:string,cid:string)

Category0 (cat_id0:string,cat_name0:string)

Category1 (cat_id1:string,cat_name1:string)

Category2 (cat_id2:string,cat_name2:string)

Resources(cid:string,rack_id:string,rtype:string,rname:string,keyword:string)

Assignment (aid:string,astart:date,aend:date,ano:integer)

Impact (cname:string,II:real)

Trend (cname:string,TI:real)

Rating(regid:integer, [cid:integer](#), score:integer, content:string)

Bidding(regid:integer, [cid:integer](#), ctime:date, price:real)

Browsing(regid:integer, keyword:string)

Schema Overview:

Forum: This table contains information about forums.

```
mysql> select * from forum;
+-----+-----+
| forum_id | forum_name |
+-----+-----+
| 1 | Week 1 Discussion |
| 2 | Week 2 Discussion |
| 3 | Week 3 Discussion |
| 4 | Week 4 Discussion |
| 5 | Week 5 Discussion |
| 6 | Cloud Computing in the News |
+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Course forum: This table contains information about the course and the respective forums.

```
mysql> select * from course_forum;
+-----+-----+
| course_id | forum_id |
+-----+-----+
| 1023 | 1 |
| 1023 | 2 |
| 1024 | 3 |
| 1024 | 4 |
+-----+-----+
4 rows in set (0.00 sec)
```

Questionnaire: This table contains information about questions which were asked in respective forums.

```
mysql> select * from questionnaire;
```

question_id	query
1	Why Naive Bayes Classifier cant be used to categorise things into multiple cat?
2	When it will be possible to register for the previous courses?
3	What is ntp
4	The grading for Question 23 in final exam is wrong?
5	Mastery Badges not handed out for last course?
6	Question Regarding Earning a Certificate?
7	Another certificate payment extension?
8	What is gossip why do we need it in cloud?
9	Is AWS uses gossip?
10	I did all the programming assignments : when do we get the certificate?
11	Availability of course content in the future?
12	Very late joining?
13	Some good, some bad for the first offering?
14	Programming track - do we need to complete all assignments?
15	Programming Track- Mastery Badges?
16	What is ShortestPath exercise?
17	What is Right Graph Dataset Format?
18	Incomplete Instruction MP5 C: Shortest Path?
19	About source code walk through videos?
20	Upon completion of all MP, when will certificate be available?
21	Yet another issue with the submit script - TopPopularLinks?
22	Next course on the cloud specialization and Capstone project?
23	What is difference between quad tree and interval tree why do we need then?

```
23 rows in set (0.00 sec)

mysql>
```

Forum questionnaire: This table contains information about the forum and the corresponding question details.

```
mysql> show tables;
+-----+
| Tables_in_educat |
+-----+
| assignments_report |
| course_forum       |
| courses            |
| e_assignments      |
| forum              |
| forum_questionnaire |
| homework           |
| homework_report    |
| question_answers   |
| questionnaire      |
| students           |
+-----+
11 rows in set (0.00 sec)

mysql> select * from forum_questionnaire;
+-----+-----+
| forum_id | question_id |
+-----+-----+
| 1       | 1          |
| 1       | 2          |
| 1       | 3          |
| 1       | 4          |
| 1       | 5          |
| 2       | 6          |
| 2       | 7          |
| 2       | 8          |
| 2       | 9          |
| 2       | 10         |
| 3       | 18         |
| 3       | 19         |
| 3       | 20         |
| 3       | 21         |
| 3       | 22         |
+-----+-----+
15 rows in set (0.00 sec)

mysql>
```

Question answers: This table contains information about the question and respective answers which are posted by different set of students.


```
mysql> select * from question_answers;
```

question_id	answer
1	naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes theorem with strong (naive) independence assumptions betw
2	Spring 2016
3	Network Time Protocol (NTP) is a networking protocol for clock synchronization between computer systems over packet-switched networks
4	Yep it is wrong
5	They handled it yesterday
6	You have to score atleast 65% in all the exams
7	Yes they extended it once again, thats a good news
8	The term epidemic protocol is sometimes used as a synonym for a gossip protocol, because gossip spreads information in a manner similar to the spread of a
9	To detect failures AWS uses gossip
10	Atleast we need 10 days to process your grades

```
10 rows in set (0.00 sec)

mysql>
```

Homework: This table tracks the home work grade for a particular student in a particular course.

```
mysql> select * from homework;
```

ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 533
Current database: educat

course_id	student_id	homework_id	name	grade	total
1023	10	1	hw1	80	100
1023	10	2	hw2	60	100
1023	10	3	hw3	90	100
1023	10	4	hw4	80	100
1023	10	5	hw5	100	100
1023	20	1	hw1	80	100
1023	20	2	hw2	70	100
1023	20	3	hw3	90	100
1023	20	4	hw4	100	100
1023	20	5	hw5	70	100
1023	30	1	hw1	80	100
1023	30	2	hw2	90	100
1023	30	3	hw3	100	100
1023	30	4	hw4	90	100
1023	30	5	hw5	100	100
1023	40	1	hw1	80	100
1023	40	2	hw2	90	100
1023	40	3	hw3	100	100
1023	40	4	hw4	90	100
1023	40	5	hw5	100	100
1023	50	1	hw1	30	100
1023	50	2	hw2	50	100
1023	50	3	hw3	10	100
1023	50	4	hw4	90	100
1023	50	5	hw5	30	100
1023	60	1	hw1	30	100
1023	60	2	hw2	50	100
1023	60	3	hw3	10	100
1023	60	4	hw4	90	100
1023	60	5	hw5	30	100
1024	10	1	hw1	80	100
1024	10	2	hw2	60	100
1024	10	3	hw3	90	100
1024	10	4	hw4	80	100
1024	10	5	hw5	100	100
1024	20	1	hw1	80	100
1024	20	2	hw2	70	100
1024	20	3	hw3	90	100
1024	20	4	hw4	100	100
1024	20	5	hw5	70	100
1024	30	1	hw1	80	100
1024	30	2	hw2	90	100

Assignments: This table tracks the assignment grade for a particular student in a particular course.

```
mysql> select * from e_assignments;
```

course_id	student_id	assignment_id	name	grade	total
1023	10	1	Assignment1	80	100
1023	10	2	Assignment2	60	100
1023	10	3	Assignment3	90	100
1023	10	4	Assignment4	80	100
1023	10	5	Assignment5	100	100
1023	20	1	Assignment1	80	100
1023	20	2	Assignment2	70	100
1023	20	3	Assignment3	90	100
1023	20	4	Assignment4	100	100
1023	20	5	Assignment5	70	100
1023	30	1	Assignment1	80	100
1023	30	2	Assignment2	90	100
1023	30	3	Assignment3	100	100
1023	30	4	Assignment4	90	100
1023	30	5	Assignment5	100	100
1023	40	1	Assignment1	80	100
1023	40	2	Assignment2	90	100
1023	40	3	Assignment3	100	100
1023	40	4	Assignment4	90	100
1023	40	5	Assignment5	100	100
1023	50	1	Assignment1	30	100
1023	50	2	Assignment2	50	100
1023	50	3	Assignment3	10	100
1023	50	4	Assignment4	90	100
1023	50	5	Assignment5	30	100
1023	60	1	Assignment1	30	100
1023	60	2	Assignment2	50	100
1023	60	3	Assignment3	10	100
1023	60	4	Assignment4	90	100
1023	60	5	Assignment5	30	100
1024	10	1	Assignment1	80	100
1024	10	2	Assignment2	60	100
1024	10	3	Assignment3	90	100
1024	10	4	Assignment4	80	100
1024	10	5	Assignment5	100	100
1024	20	1	Assignment1	80	100
1024	20	2	Assignment2	70	100
1024	20	3	Assignment3	90	100
1024	20	4	Assignment4	100	100
1024	20	5	Assignment5	70	100
1024	30	1	Assignment1	80	100
1024	30	2	Assignment2	90	100
1024	30	3	Assignment3	100	100
1024	30	4	Assignment4	90	100
1024	30	5	Assignment5	100	100

Reporting:

All reporting is handled through views. For every report either we will create a new view or we will use an existing view to generate that report.

Example:-

Sample Scripts

create or replace view homework_report as select course_id,student_id,sum(grade) as h_grade,sum(total) as h_total from homework group by course_id,student_id;

For full set of reporting please refer to our git hub account.

Cumulative homework report

```
mysql> select * from homework_report;
```

course_id	student_id	h_grade	h_total
1023	10	410	500
1023	20	410	500
1023	30	460	500
1023	40	460	500
1023	50	210	500
1023	60	210	500
1024	10	410	500
1024	20	410	500
1024	30	460	500
1024	40	460	500
1024	50	210	500
1024	60	290	500

```
12 rows in set (0.00 sec)
```

Cumulative assignment report

```
mysql> select * from assignments_report;
```

course_id	student_id	a_grade	a_total
1023	10	410	500
1023	20	410	500
1023	30	460	500
1023	40	460	500
1023	50	210	500
1023	60	210	500
1024	10	410	500
1024	20	410	500
1024	30	460	500
1024	40	460	500
1024	50	210	500
1024	60	290	500

```
12 rows in set (0.00 sec)
```

```
mysql>
```

5. SQL STATEMENTS

4.1. SQL TABLE Creation and Loading data: - Core Model

```
create table courses(cid char(20),cname char(30),primary key(cid));
```

```
create table lecturers(lid char(20),uid char(20),primary key(lid));
```

```
create table professor_course(pid char(20),cid char(20),primary key(pid),foreign key(pid)  
references lecturers(lid),foreign key(cid) references courses(cid));
```

```
create table students(sid char(20),sname char(50),primary key(sid));
```

```
create table student_course(sid char(20),cid char(20),primary key(sid),foreign key(sid)  
references students(sid),foreign key(cid) references courses(cid));
```

```
create table category(cid char(20),cat_id0 char(20),cat_id1 char(20),cat_id2 char(20) ,primary  
key(cid,cat_id0,cat_id1,cat_id2),foreign key(cid) references courses(cid));
```

```
create table category0(cat_id0 char(20),cat_name0 char(50),primary key (cat_id0));
```

```
create table category1(cat_id1 char(20),cat_name1 char(50),primary key (cat_id1));
```

```
create table category2(cat_id2 char(20),cat_name2 char(50),primary key (cat_id2));
```

```
--
```

```
create table assignments(cid char(20),aid char(20),astart date,aend date,primary  
key(cid,aid),foreign key(cid) references courses(cid) on delete cascade);
```

```
create table resources(cid char(20),rack_id char(20),rtype char(20),rname char(20),keyword  
char(20),primary key(cid,rack_id),foreign key(cid) references courses(cid)on delete cascade);
```

```
create table impact(cname char(30),ii real,primary key(cname));
```

```
create table trend(cname char(30),ti real,primary key(cname));
```

```
create table Rating(regid int(20),cid char(20),score integer, content char(255),primary key(regid,  
cid), foreign key(regid) references registered_user(regid) on delete cascade, foreign key(cid)  
references courses(cid)on delete cascade);
```

create table Bidding(regid int(20),cid char(20),ctime date, price real,primary key(regid, cid), foreign key(regid) references registered_user(regid) on delete cascade, foreign key(cid) references courses(cid)on delete cascade);

create table Browsing(regid int(20),keyword char(20),primary key(regid), foreign key(regid) references registered_user(regid) on delete cascade,);

create table forum(forum_id int(20),forum_name char(200),primary key(forum_id));

create table questionnaire(question_id int(20), query char(200),primary key(question_id));

*create table forum_questionnaire(forum_id int(20),
question_id int(20),
foreign key(forum_id) references
forum(forum_id) on delete cascade,
foreign key(question_id) references
questionnaire(question_id) on delete cascade);*

create table course_forum(course_id int(20),forum_id int(20),primary key(course_id,forum_id),foreign key(course_id) references courses(cid) on delete cascade,foreign key(forum_id) references forum(forum_id) on delete cascade);

create table question_answers(question_id int(20), answer char(200) ,student_id int(20), primary key(question_id,student_id),foreign key(question_id) references questionnaire(question_id), foreign key(student_id) references students(sid) on delete cascade);

create table homework(course_id int(20), student_id int(20), homework_id int(20), name char(200),grade int(10),total int(10), primary key(course_id,student_id,homework_id),foreign key(course_id) references courses(cid) on delete cascade,foreign key(student_id) references students(sid) on delete cascade);

create table e_assignments(course_id int(20), student_id int(20), assignment_id int(20), name char(200),grade int(10),total int(10),primary key(course_id,student_id,assignment_id),foreign key(course_id) references courses(cid) on delete cascade,foreign key(student_id) references students(sid) on delete cascade);

create table registered_user(regid char(20), firstname varchar(30), middlename varchar(30), lastname varchar(30), dob date, email varchar(50), pass varchar(512), regdate date, primary key(regid), unique(email));

create table students(sid char(20), type varchar(20), profile0 varchar(512), profile1 varchar(512), profile2 varchar(512), area0 varchar(40), area1 varchar(40), area2 varchar(40), primary key(sid), foreign key(sid) references registered_user(regid) on delete cascade);

create table universities(uname varchar(50) not null, uwebpage varchar(512), primary key(uname));

create table lecturers(lid char(20), type varchar(20), uname varchar(50), area0 varchar(40), area1 varchar(40), area2 varchar(40), primary key(lid), foreign key(lid) references registered_user(regid) on delete cascade, foreign key(uname) references universities(uname) on delete no action);

create table certificates(certid char(20), ctype varchar(50), cgrade varchar(5), clevel int(20), primary key(certid));

create table courseschedule(cid char(20), fromdate date, enddate date, foreign key(cid) references courses(cid) on delete cascade, primary key(cid));

create table certification(certindex char(20), sid char(20), cid char(20), issuedate date, certid char(20), foreign key(sid) references students(sid), foreign key(cid) references courses(cid), foreign key(certid) references certificates(certid), primary key(certindex));

Sample population scripts are posted in github.

6. Technology Survey

5.1. DESIGN APPROACH

In phase 2, we use the ER diagram in the phase 1 to know that what schemas do we need. Then we build the schemas first. According to ER diagram, we know the key constraint of schemas. Second, we state the FDs of schemas. According to FDs, we normalize it into BCNF. Then we create the tables and populate the data.

In phase 3, we will build the system according to the database. We can divide our system into 3 sides: front side, server side and database. We need to develop server side to achieve the logic and computation of the system, and to call the data in database. We need to develop front side to show the result of data to users. Combine the 3 sides, our system is finished.

5.2. TECHNOLOGY

Webpages	HTML5/JavaScript/CSS
Server side	Apache + PHP
Database	MySQL
Database IDE	PHPMysqlAdmin
Operating System	Windows

Technology Stack	WAMP(Windows, Apache, MySQL, PHP)
------------------	-----------------------------------

Totally, we implement our system as a website not local software system, because the website is more lightweight and convenient to show our works. We need to decide which technology we use to build the website.

Webpages: The most popular webpage technology – HTML5/JavaScript/CSS is our choice. This technology has high performance and strong functions. It also has a good compatibility. What's more, it is easy for our team members to learn. So we choose it.

Server side: In the server side, we use PHP language. The first reason is that PHP is suitable to develop a simple website. Because of its lightweight framework, PHP will save more time than other programming language such as JAVA (JAVA is famous for its complexity of developing a simple website). PHP is descriptive language, it has a good compatibility, and it's easy to learn. This is also an important reason.

Database: Of course, we will use SQL to build the database. The point is that we choose MySQL or SQL server. Because SQL server is Microsoft's and it's a little big for a simple project. So we choose MySQL.

Database IDE: Database IDE is also important. A good IDE can improve our efficiency. PHP has many mature frameworks and they are helpful. We choose PHPMysqlAdmin to manage our database.

7. POPULATE DATA FOR DATABASE

6.1. CORE MODEL

Table Name: Courses

cid	cname
1000	Basic VLSI
1001	BasicDBMS
1002	BasicMicro Architecture
1003	BasicOrthopaedics_best_practices
1004	BasicMRI
1005	BasicHomicides
1006	BasicDetective training
1007	BasicReal estate law
1008	BasicConsumer law
1009	BasicMarket volatility
1010	BasicBusiness Administration
1011	BasicRoman Architecture
1012	BasicDats Structures and Algorithms

Fig 14. Sample entries in Table Courses

Table Name: professor_course

pid	cid
2000	1000
2001	1001
2002	1002
2003	1003
2004	1004
2005	1005

Fig 15. Sample entries in Table professor_course

Table Name: student_course

sid	cid
3000	1029
3001	1039
3002	1028
3003	1016
3004	1006

Fig 16.Sample entries in Table student_course

Table Name: Category

cid	cat_id0	cat_id1	cat_id2
1000	101	203	301
1001	101	201	302
1002	101	201	303
1003	102	205	304
1004	102	202	305
1005	103	206	306

Fig 17.Sample entries in Table Category

Table Name: Category0,Category1,Category2(Hierarchical)

cat_id0	cat_name0	cat_id1	cat_name1	cat_id2	cat_name2
101	Engineering	201	Computer Science Engineering	301	VLSI
102	Medicine	202	Radiology	302	DBMS
103	Law	203	Electrical Engineering	303	Micro Architecture
104	Business	204	Finance	304	Orthopaedics_best_practices
105	Arts and Human Sciencce	205	Orthopaedics	305	MRI

Fig 18.Sample entries for Category tree

Table Name: Assignments

cid	aid	astart	aend
1029	1	2015-10-15	2015-10-17
1039	2	2015-10-16	2015-10-18
1028	1	2015-10-20	2015-10-22
1016	5	2015-10-22	2015-10-24
1006	4	2015-10-23	2015-10-26
1021	2	2015-11-04	2015-11-08

Fig 19.Sample entries in Table Assignments

Table Name: Resources

cid	rack_id	rtype	rname	keyword
1031	9763	ebook	R1	r
1015	9122	video	R2	p
1021	9593	html	R3	e
1009	9123	ebook	R4	v
1044	9905	video	R5	a

Fig 20.Sample entries in Table Resources

Table Name: Impact & Trend

cname	TI	cname	II
Basic VLSI	36.4	Basic VLSI	39.9
BasicDBMS	77.8	BasicDBMS	72.6
BasicMicro Architecture	53.9	BasicMicro Architecture	66.5
BasicOrthopaedics_best_practices	25.9	BasicOrthopaedics_best_practices	22.4
BasicMRI	32.2	BasicMRI	92.6

Fig 21.Sample table in entries in Impact and Trend respectively

6.2. HR MODEL

Tables in database and sample data

```
mysql> select * from registered_user;
```

regid	firstname	middlename	lastname	dob	email	pass	regdate
2003	Otto	Gunther	Octavius	1910-10-10	dr.octopus@gmail.com	@#\$23DF%\$%e454\$%\$W#QR	2014-07-09
2004	Bruce		Banner	1937-06-09	hulk.smash@gmail.com	@#\$23DF%#\$DES@#RFedd	2012-03-03
3000	Pawan	Kumar	Murty	1901-10-10	wolverine007@gmail.com	rwci0jflkspt4rj#%\$#%	2015-10-12
3001	Peter		Parker	1947-10-10	spidy.friendlynbd@gmail.com	@#\$#dsfe454\$%\$W#QR	2015-10-06
3002	Tony		Stark	1929-10-10	ironman.gbpp@gmail.com	@#\$#dsfe454\$%\$W#QR	2014-12-06

5 rows in set (0.00 sec)

```
mysql> select *from students;
```

sid	type	profile0	profile1	profile2	area0	area1	area2
3000	Student	http://linkedin.com/pawan.murty			Database and Managment Systems	Genetics	Mutation
3001	Professional	http://linkedin.com/spidy.friendlynbd			Computer Architecture	Database and Management Systems	
3002	Self-Employed	http://linkedin.com/ironman.gbpp			Computer Vision	Database and Management Systems	Money Manage

3 rows in set (0.00 sec)

```
mysql> select *from lecturers;
```

lid	type	uname	area0	area1	area2
2003	Professional	Penn State UP	Genetics	Mutation	Neural Networks
2004	Part-Time	NIT Warangal	Anger Management	Astrophysics	

2 rows in set (0.00 sec)

```
mysql> select *from universities;
```

uname	uwebpage
IIT Kanpur	http://iitk.ac.in
NIT Warangal	http://nitw.ac.in
Penn State UP	http://psu.edu
University of Texas, Austin	http://www.utexas.edu

4 rows in set (0.00 sec)

Fig 22. SQL snapshot showing sample databases.

```
mysql> select *from courseschedule;
+-----+-----+-----+
| cid | fromdate | enddate |
+-----+-----+-----+
| 1000 | 2015-06-10 | 2016-12-10 |
| 1001 | 2015-06-13 | 2016-12-10 |
| 1002 | 2015-01-24 | 2016-06-24 |
| 1010 | 2015-01-01 | 2016-06-01 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select *from certificates;
+-----+-----+-----+-----+
| certid | ctype | cgrade | clevel |
+-----+-----+-----+-----+
| 5000 | Student | Distinction | Expert |
| 5001 | Professional | First Class | Beginner |
| 5002 | Student | Second Class | Advanced |
| 5003 | Student | Distinction | Beginner |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select *from certification;
+-----+-----+-----+-----+-----+
| certindex | sid | cid | issuedate | certid |
+-----+-----+-----+-----+-----+
| 6000 | 3000 | 1000 | 2015-09-08 | 5000 |
| 6001 | 3002 | 1001 | 2015-09-08 | 5002 |
| 6002 | 3002 | 1010 | 2015-07-01 | 5001 |
| 6003 | 3001 | 1003 | 2015-06-09 | 5003 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Fig 23. SQL Snapshot showing sample Databases

6.3. PR MODEL

Table Name: Forum

`SELECT * FROM `forum``

Number of rows: 25

Sort by key: None

+ Options

		forum_id	forum_name
<input type="checkbox"/>	Edit Copy Delete	1	Week 1 Discussion
<input type="checkbox"/>	Edit Copy Delete	2	Week 2 Discussion
<input type="checkbox"/>	Edit Copy Delete	3	Week 3 Discussion
<input type="checkbox"/>	Edit Copy Delete	4	Week 4 Discussion
<input type="checkbox"/>	Edit Copy Delete	5	Week 5 Discussion
<input type="checkbox"/>	Edit Copy Delete	6	Cloud Computing in the News

↑ ☐ Check All With selected: Change Delete Export

Number of rows: 25

Fig 22: Sample entries in forum table

Table Name: Course Forum

`SELECT * FROM `course_forum``

Number of rows: 25

Sort by key: None

+ Options

		course_id	forum_id
<input type="checkbox"/>	Edit Copy Delete	1023	1
<input type="checkbox"/>	Edit Copy Delete	1023	2
<input type="checkbox"/>	Edit Copy Delete	1024	3
<input type="checkbox"/>	Edit Copy Delete	1024	4

↑ ☐ Check All With selected: Change Delete Export

Number of rows: 25

Fig 23: Sample entries in course forum table

Table Name: Questionnaire

<pre>SELECT * FROM `questionnaire`</pre>	
Number of rows:	25
Sort by key:	None
+ Options	
← T →	question_id query
<input type="checkbox"/> Edit Copy Delete	1 Why Naive Bayes Classifier cant be used to categor...
<input type="checkbox"/> Edit Copy Delete	2 When it will be possible to register for the previ...
<input type="checkbox"/> Edit Copy Delete	3 What is ntp
<input type="checkbox"/> Edit Copy Delete	4 The grading for Question 23 in final exam is wrong...
<input type="checkbox"/> Edit Copy Delete	5 Mastery Badges not handed out for last course?
<input type="checkbox"/> Edit Copy Delete	6 Question Regarding Earning a Certificate?
<input type="checkbox"/> Edit Copy Delete	7 Another certificate payment extension?
<input type="checkbox"/> Edit Copy Delete	8 What is gossip why do we need it in cloud?
<input type="checkbox"/> Edit Copy Delete	9 Is AWS uses gossip?
<input type="checkbox"/> Edit Copy Delete	10 I did all the programming assignments : when do we...
<input type="checkbox"/> Edit Copy Delete	11 Availability of course content in the future?
<input type="checkbox"/> Edit Copy Delete	12 Very late joining?
<input type="checkbox"/> Edit Copy Delete	13 Some good, some bad for the first offering?
<input type="checkbox"/> Edit Copy Delete	14 Programming track - do we need to complete all ass...
<input type="checkbox"/> Edit Copy Delete	15 Programming Track- Mastery Badges?

Fig 24: Sample table entries in Questionnaire table

`SELECT * FROM `question_answers``

Number of rows: 25

Sort by key: None

+ Options

	question_id	answer	student_id
<input type="checkbox"/> Edit Copy Delete	1	naive Bayes classifiers are a family of simple pro...	10
<input type="checkbox"/> Edit Copy Delete	2	Spring 2016	10
<input type="checkbox"/> Edit Copy Delete	3	Network Time Protocol (NTP) is a networking protoc...	10
<input type="checkbox"/> Edit Copy Delete	4	Yup it is wrong	10
<input type="checkbox"/> Edit Copy Delete	5	They handled it yesterday	10
<input type="checkbox"/> Edit Copy Delete	6	You have to score atleast 65% in all the exams	10
<input type="checkbox"/> Edit Copy Delete	7	Yes they extended it once again, thats a good news	10
<input type="checkbox"/> Edit Copy Delete	8	The term epidemic protocol is sometimes used as a ...	10
<input type="checkbox"/> Edit Copy Delete	9	To detect failures AWS uses gossip	10
<input type="checkbox"/> Edit Copy Delete	10	Atleast we need 10 days to process your grades	40

☐ Check All
 With selected:
 ☐ Change
 ☐ Delete
 ☐ Export

Fig 25: Sample table entries in Question answers table

Student Grade Reports:

All reports are based on factual data.

Here are the screen shots of some of the reports we have generated so far.

Students who failed in the homework's

✓ Showing rows 0 - 2 (3 total, Query took 0.0026 sec)

```
select mark.course_id,mark.student_id,mark.m as average_marks from (select course_id,student_id,(h_grade/h_total)*100 as m from homework_report)as mark , homework_report where mark.m < 50 and mark.student_id=homework_report.student_id and homework_report.course_id=mark.course_id
```

☐ Profiling [\[Inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP Code \]](#) [\[Refresh \]](#)

Number of rows: 25 ▾

+ Options

course_id	student_id	average_marks
1023	50	42.0000
1023	60	42.0000
1024	50	42.0000

Number of rows: 25 ▾

Fig 26: Sample Report for students who failed in the test

Students who passed in the homework's

```
select mark.course_id,mark.student_id,mark.m as average_marks from (select course_id,student_id,(h_grade/h_total)*100 as m from homework_report)as mark , homework_report where mark.m >= 50 and mark.student_id=homework_report.student_id and homework_report.course_id=mark.course_id
```

☐ Profiling [\[Inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP Code \]](#) [\[Refresh \]](#)

Number of rows: 25 ▾

+ Options

course_id	student_id	average_marks
1023	10	82.0000
1023	20	82.0000
1023	30	92.0000
1023	40	92.0000
1024	10	82.0000
1024	20	82.0000
1024	30	92.0000
1024	40	92.0000
1024	60	58.0000

Fig 27: Sample Report for students who passed in the test

Home Page:

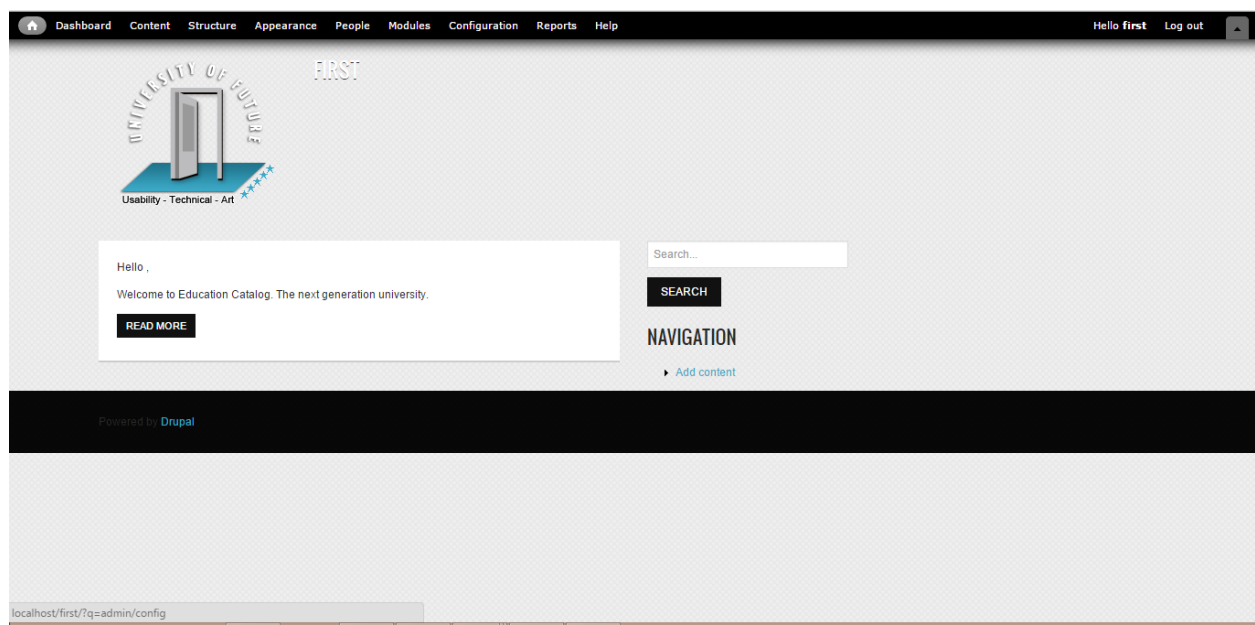


Fig 28: Home Page