**Final project DBMS – 1**

**Made by: Arina Ussubaliyeva**

**Meruyert Kerimbayeva**

**Sabina Rakhmetova**

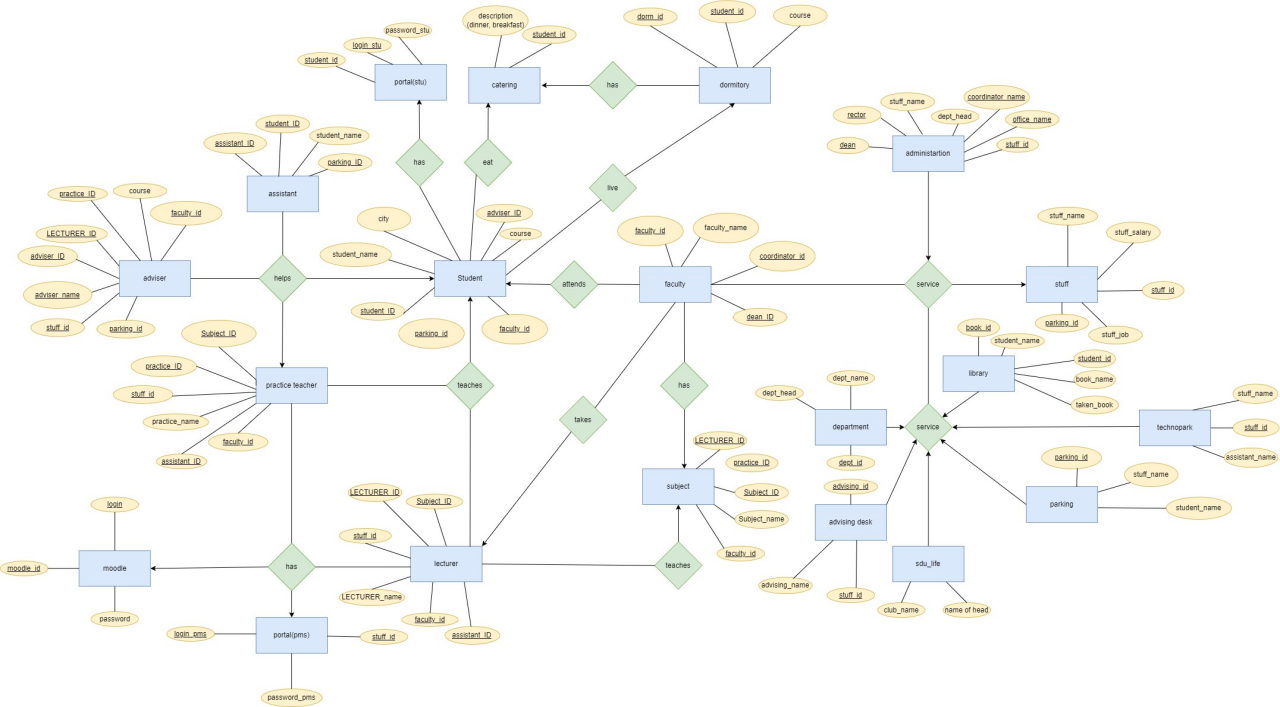
**Introduction**

This project about the university database system.Our system basically consists based on these beliefs:We have students that have their own accounts in the university portal and the same works for teacher and staff members.

Using all the information provided by the database, you can easily find out in which faculty, course and subjects the student studies and for teacher(lecturer and practice) which subjects they teach and in which faculty they work.

In addition, we have Dorm tables from where we can find out which students live there, also we have catering, which has attributes with description about dinner. Parking tables from where we can find out which staff and students have permission for parking.

1. **ER Diagram**



1. **Functional Dependencies**

**1 table** Student

Student\_ID, Student\_name, City, Course, Adviser\_ID, Faculty\_ID, Parking\_ID

**2 table** Adviser

Adviser\_ID, Adviser\_name, Course, Faculty\_ID, Lecturer\_ID, Practice\_ID, Stuff\_ID, Parking\_ID

**3 table** Practice\_Teacher

Practice\_ID, Practice\_Name, Subject\_ID, Assistant\_ID, Faculty\_ID, Stuff\_ID

**4 table** Lecturer

Lecturer\_ID, Lecturer\_Name, Subject\_ID, Assistant\_ID, Faculty\_ID, Stuff\_ID

**5 table** Subject

Subject\_ID, Subject\_name, Lecturer\_ID, Practice\_ID, Faculty\_ID

**6 table** Faculty

Faculty\_ID, Faculty\_Name, Coordinator\_ID, Dean\_ID

**7 table** Assistant

Assistant\_ID, Student\_ID, Student\_name, Parking\_ID

**8 table** Administration

Stuff\_ID, Stuff\_name, Coordinator\_name, Office\_name, Dept\_head, Rector, Dean

**9 table** Department

Dept\_ID, Dept\_name, Dept\_head

**10 table** Library

Book\_ID, Book\_name, Taken\_Book, Student\_ID, Student\_name

**11 table** Stuff

Stuff\_ID, Stuff\_name, Stuff\_salary, Stuff\_job, Parking\_ID

**12 table** Technopark

Stuff\_ID, Stuff\_name, Assistant\_Name

**13 table** SDU\_life

Club\_name, Club\_head

**14 table** Advising\_Desk

Advising\_ID, Advising\_name, Stuff\_ID

**15 table** Parking

Parking\_ID, Student\_name, Stuff\_Name

**16 table** Dormitory

Dorm\_ID, Student\_ID, Course

**17 table** Catering

Student\_ID, Description

**18 table** Portal(stu)

Student\_ID, Login\_stu, Password\_stu

**19 table** Portal(pms)

Stuff\_ID, Login\_pms, Password\_pms

**20 table** Moodle

Moodle\_ID, Login, Password

1. **Normalization**

**1NF**

All tables in 1NF, because all attributes of these relations are simple.

**2NF**

All table are in 2NF, because they are in 1NF and have primary keys : Student\_ID, Adviser\_ID, Practice\_ID, Lecturer\_ID, Stuff\_ID, Parking\_ID, Moodle\_ID, Subject\_ID, Faculty\_ID, Dept\_ID, Advising\_ID, Book\_ID, Dorm\_ID, Assistant\_ID

**3NF**

Tables Student, Catering, Adviser, Assistant, Practice\_teacher, Lecturer, Moodle, Portal, Portal(pms),

Subject, Faculty, Department, Advising\_desk, Library, Stuff, Parking

are in 3NF, because they are reduced to 2NF. and all non-key attributes are mutually independent and completely dependent on the key.

**BCNF**

Some tables are in BCNF

1. **Queries**

1) SELECT Student\_ID, Description

FROM Catering;

2) SELECT Student\_name, city

FROM Student

WHERE city= ‘Kuala Lumpur’ OR ‘Oslo’;

3) INSERT into Portal\_pms

VALUES (9100, ‘coco@jimbo.com’, ‘fFfFfF’);

4) UPDATE Practice\_teacher

SET Practice\_name= ‘Mutaliev Nauryzbay’

WHERE stuff\_id = 1281;

5) Alter table Student

ADD Student\_Gender VARCHAR(6);

6) DELETE

FROM library

WHERE Student\_ID = 880;

7) SELECT head\_name , club\_name

FROM sdu\_life

ORDER BY club\_name asc;

8) SELECT stuff.stuff\_id, Stuff.salary, Technopark.stuff\_name,

Technopark.Assistant\_name

FROM Technopark

INNER JOIN

Stuff ON

Technopark.Stuff\_ID = Stuff.Stuff\_ID;

9) SELECT MIN(salary)

FROM Stuff

WHERE parking\_id IS NOT NULL;

10) SELECT stuff\_ID

FROM adviser

INTERSECT

SELECT stuff\_ID

FROM administration

11) SELECT SUM(salary)

FROM Stuff

WHERE Stuff\_name= ‘A%’;

12) SELECT AVG(course)

FROM Student

WHERE Student\_ID > 100;

13) CREATE TRIGGER New\_Student1

AFTER UPDATE ON dormitory

WHEN new.course = 2

BEGIN

INSERT INTO dormitory (

student\_id,

dorm\_id,

course

)

VALUES

(111,

222,

1

) ;

END

14) UPDATE dormitory

SET course = 2

WHERE Student\_id = 638

15) SELECT practice\_teacher.practice\_name,

practice\_teacher.subject\_ID, lecturer.lecturer\_name, lecturer.subject\_ID

FROM practice\_teacher

LEFT JOIN

lecturer ON

lecturer.subject\_id = practice\_teacher.subject\_id;

16) SELECT Practice\_name

FROM Practice\_teacher

UNION

SELECT Lecturer\_name

FROM Lecturer

17) BEGIN TRANSACTION;

UPDATE stuff

SET salary=salary + 10000

WHERE stuff\_id= 2051;

INSERT INTO department (dept\_ID, dept\_head, dept\_name)

VALUES(7, 'Dama Lama', 'social');

COMMIT;

18) SELECT Student\_id FROM portal\_stu

Except

SELECT Stuff\_id FROM portal\_pms

19) UPDATE student

SET parking\_id = NULL

WHERE course = 1

SELECT student\_id

FROM student

WHERE parking\_id IS NULL;

20) Group by,

SELECT lecturer\_id, lecturer\_name, faculty\_id, subject\_id

FROM Lecturer

WHERE faculty\_id=004

GROUP BY Lecturer\_id

ORDER by subject\_id