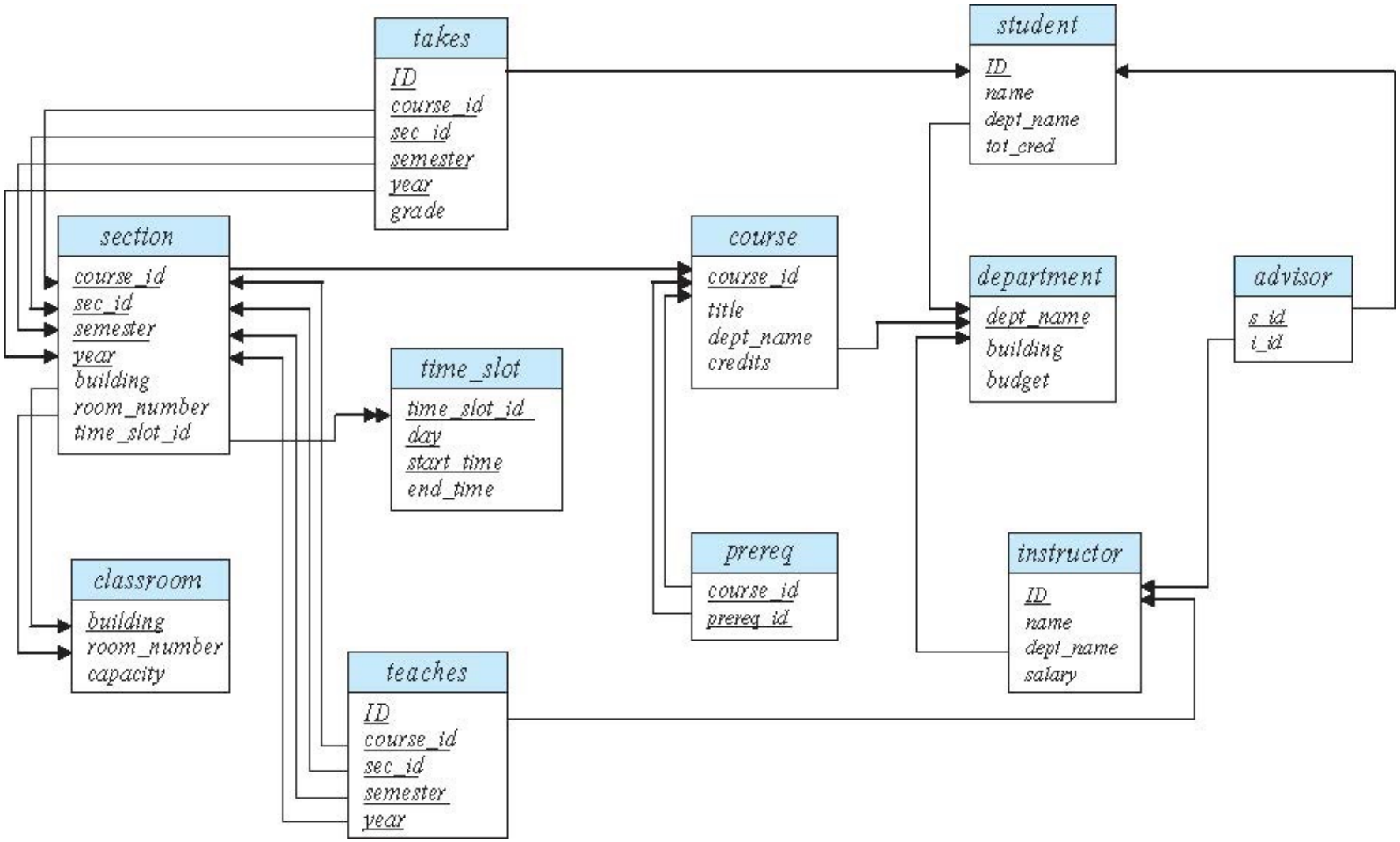
Schema Diagram for University Database



Problem 1 (demo)

There are many useful functions in SQL. They may vary slightly from one DBMS to another. They functions **now**() , **curdate** (), and **version** (). Note that these functions do not require a **from** clause.

```
select now();
select curdate ();
select version ();
```

Problem 2 (demo)

There is not only LIKE, but also REGEXP operator in SQL that allows string matching using the full power is regular expressions.

(details here https://dev.mysql.com/doc/refman/8.0/en/regexp.html#regexp-syntax)

Syntax of REGEXP operator is identical to LIKE:

<string> REGEXP <matching pattern>

Find all courses which titles contain the word “cat” or “cats”.

Answer Problem 2

Find all courses which titles contain the word “cat” or “cats”.

select distinct title **from** course **where** title **regexp** "(^|)cats*(|$)";

Problem 3 (demo)

Function substr(string, start, end) retrieves the portion of a string.

Another useful function in SQL is **concat()** that concatenates strings.

It takes a variable number arguments – strings to be concatenated.

Write a query that produce a list of students in ‘Math’ department. Output student name, ID, and email. Email accounts are generated as student name, flowered by ‘_’, followed by followed by the first three digits of ID. University name is “Georgetown”.

Answer Problem 3

Write a query that produce a list of students in ‘Math’ department. Output student name, ID, and email. Email accounts are generated as student name, flowered by ‘_’, followed by the first three digits of ID. University name is “Georgetown”.

```
select name, id, concat( name, '_', substr(id,1,3), '@georgetown.edu') as email
from student where dept_name='Math';
```

Problem 4 (demo)

- As you remember, **ENUM** data type is used in the MySQL database table to select one value from the predefined list.

- A column declared as ENUM will not accept any value outside the list. For example:

```
create table … ( …
    membership ENUM('Silver', 'Gold', 'Diamond'),
…);
```

- The **AUTO_INCREMENT** attribute can be used to generate a unique identity for new rows:

```
create table …(
    id  int  AUTO_INCREMENT primary key,
    … );
```

if the value of id is not defined in the insert statement, a unique value for id is generated.

a. Create a table stdprog (id, student_name, program_name) that stores programs to which students are accepted. Program name can only be ‘BS’, ‘MS’, or ‘PhD’.

b. Insert a record with your name and ‘MS’ as your program.

c. Insert a record with a different name and ‘PostDoc’ program. Observe what happens.

Answer to Problem 4

Create a table stdprog (id, student_name, program_name) that stores programs to which students are accepted. Program name can only be ‘BS’, ‘MS’, or ‘PhD’.

Insert a record with your name and ‘MS’ as your program.

Insert a record with a different name and ‘PostDoc’ program. Observe what happens.

```
create table stdprog (
id int auto_increment primary key,
name varchar(12),
program ENUM('BS', 'MS', 'PhD'));

insert into stdprog (name, program) value (‘Irina','MS');
insert into stdprog (name, program) value (‘Amy', ‘PostDoc');
select * from stdprog;
drop table stdprog;
```

Problem 5A (to submit)

Write a query that calculates number of students that received each possible grade per each course, each year. The query should provide count summaries by year by course_id, and also by year totals.

Problem 5B (to submit)

Modify the previous query to output meaningful labels instead if NULLs, You can use functions **if**() and **grouping**().