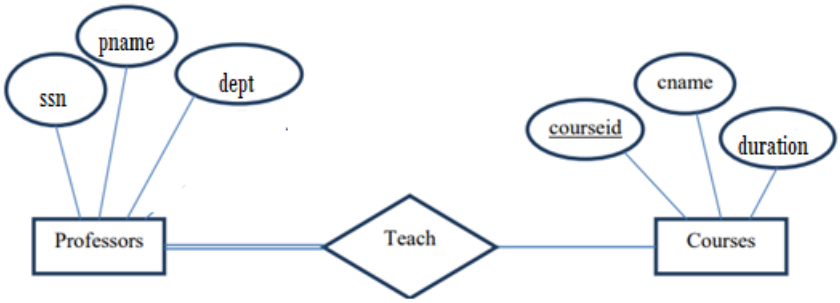
Exercise 1

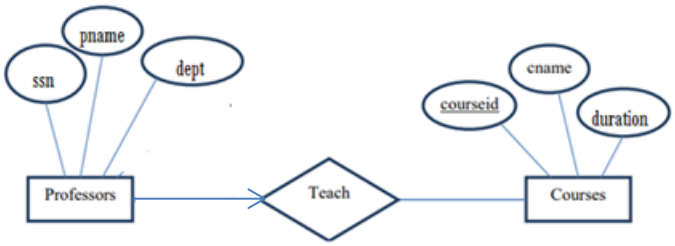
A university DB contains information about professors (identified by SIN) and courses (identified by course ID). Professors teach courses; each of the following situations concerns the Teaches relationship set.

List all candidate keys of the Teaches relationship set. a. Professors can teach the same course in several semesters, and each offering must be recorded. b. Professors can teach the same course in several semesters, but only the most recent such offering needs to be records. Assume the above Situation (b) applies in all subsequent situations.
List all the keys possible in each of the following situations.

a. Every professor teaches a course, and every course is taught by some professor.



b. Every professor teaches exactly one course, and every course is taught by exactly one professor.



MySQL 8.0 Command Line Client

```
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| semister_id       | varchar(10) | NO   | PRI | NULL    |       |
| semister_duration | varchar(50) | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> desc professors;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| Professors_Sin | varchar(20) | NO   | PRI | NULL    |       |
| ProfessorsName | varchar(50) | YES  |     | NULL    |       |
| ProfessorsDept | varchar(30) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> desc course;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| CourseId       | varchar(20) | NO   | PRI | NULL    |       |
| CourseName     | varchar(20) | YES  |     | NULL    |       |
| CourseDuration | varchar(30) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql>
```

a) Entity sets - professor: with SIN underlined as the primary key, - course: with CID underlined as the primary key, - semester: with SID underlined as the primary key.

Relationship set - teaches : associates professor, course and semester. No other attributes. The cardinality constraint is m-to-m. There is a single candidate key of the teaches relationship: {SIN, CID, SID}. The participation constraint can be anything; let say that it is total on professor and course entity sets.

Semester does not need to be an entity set here. Teaches is a binary relation between professor and course. Semester is attribute of teaches. The key o f teaches is {SIN, CID}. 1c) This means total participation from professors and total participation from courses. Because it is still m-to-m, the candidate key remains {SIN, CID}. 1d) This time the relationship is 1-to-1. There are now two candidate keys: either {SIN} or {CID}.

2a) Entity sets - employee : with SIN as the primary key; - department with DID as the key. Relationship set - works-in : with the attribute interval. m-to-m relationship with the candidate key being {SIN, DID}. 2b) Need another e ntity set for intervals (invent your own key). works-in will be a ternary relationship with the candidate key being {SIN, DID, interval's key}. 3) For Student: sID is a superkey, candidate key and

primary key. Any superset of sID is a superkey. For Course: cID is a superkey, candidate key and primary key. Any superset of cID is a superkey. Enrolled ia a m-to-m relationship. {sID, cID} is the candidate key for Enrolled, and its primary key. Both {sID, cID} and {sID, cID, grad} are superkeys.