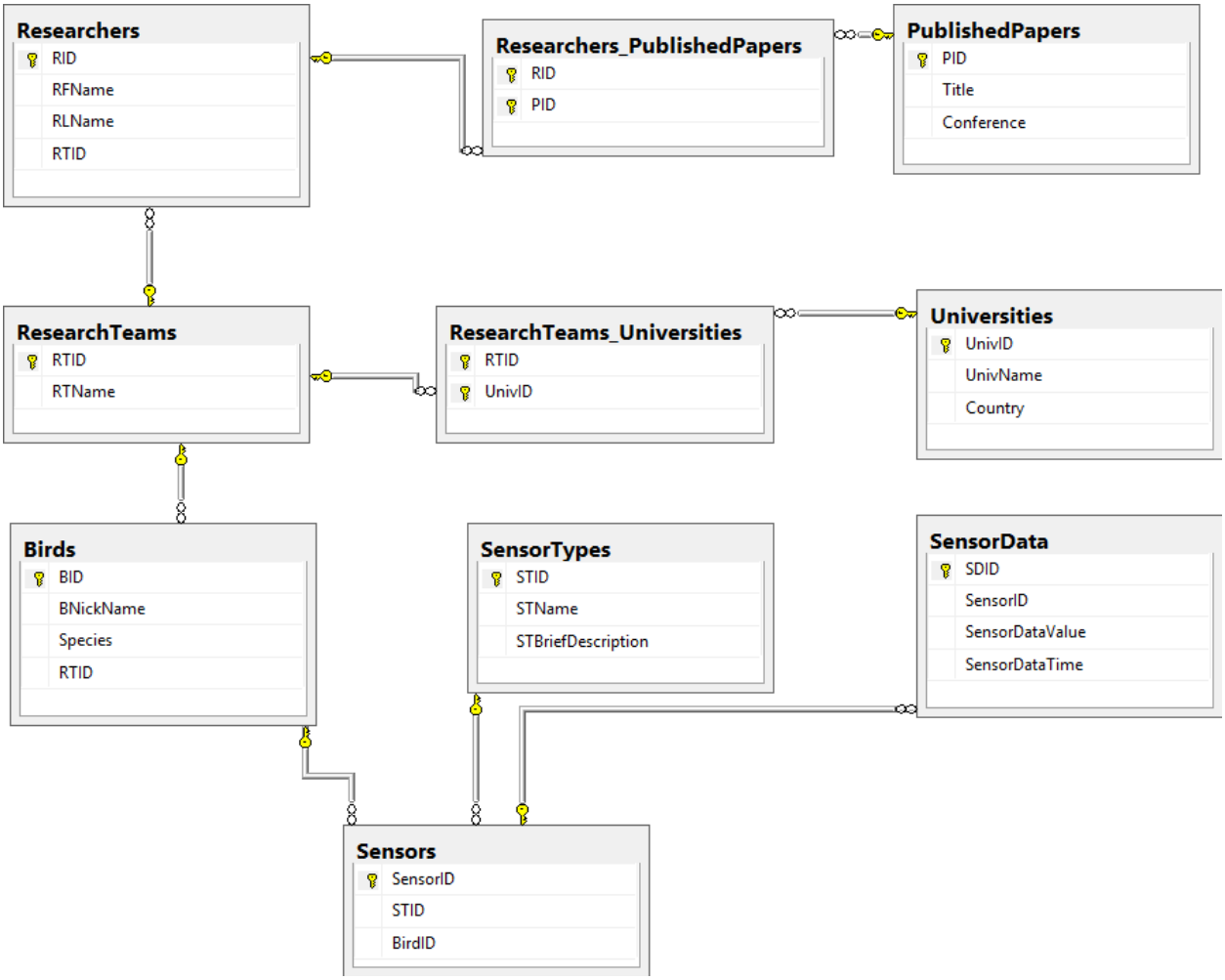
Relational database design

SQL - DDL Commands

1. Relational DB design



2. SQL DDL commands

The following command creates the *Students* table (relation). Observe that the type (domain) of each field (attribute) is specified, and enforced by the DBMS whenever tuples are added or modified.

```
CREATE TABLE Students
       (sid CHAR(20),
       name CHAR(50),
       email CHAR(30),
       age INTEGER,
```

```
      gr INTEGER)
```

As another example, the *Enrolled* table holds information about courses that students take.

```
CREATE TABLE Enrolled
      (sid CHAR(20),
      cid CHAR(5),
      grade REAL)
```

To destroy (remove) the relation *Students* the following command could be used. Of course, both schema information and the tuples are deleted.

```
DROP TABLE Students
```

Using the following command, the schema of *Students* table is altered by adding a new field; every tuple in the current instance is extended with a *null* value in the new field.

```
ALTER TABLE Students
ADD firstYear INTEGER
```

SQL could be used to declare many *candidate keys* (specified using UNIQUE), one of which is chosen as the *primary key*.

Sample: For a given student and course, there is a single grade.

```
CREATE TABLE Enrolled
(sid CHAR(20),
cid CHAR(20),
grade CHAR(2),
PRIMARY KEY (sid,cid))
```

Sample: *Students can take only one course, and receive a single grade for that course; further, no two students in a course receive the same grade*. (this is an example about how to not define candidate keys; used carelessly, an IC can prevent the storage of database instances that arise in practice!)

```
CREATE TABLE Enrolled
      (sid CHAR(20),
      cid CHAR(20),
      grade CHAR(2),
      PRIMARY KEY(sid),
```

```
      UNIQUE (cid, grade))
```

Sample of defining foreign keys "*Only students listed in the Students relation should be allowed to enroll for courses*".

```
CREATE TABLE Enrolled
      (sid CHAR(20),
      cid CHAR(20),
      grade CHAR(2),
      PRIMARY KEY (sid,cid),
      FOREIGN KEY (sid) REFERENCES Students )
```

Referential Integrity

Starting with SQL-99 there is support for all 4 approaches preserving referential integrity in case of deletes and updates:

- NO ACTION (delete/update is rejected) – it is the default approach is
- CASCADE (also delete all tuples that refer to deleted tuple)
- SET NULL/SET DEFAULT (sets foreign key value of referencing tuple)

```
CREATE TABLE Enrolled
     (sid CHAR(20),
     cid CHAR(20),
     grade CHAR(2),
     PRIMARY KEY (sid,cid),
     FOREIGN KEY (sid)
         REFERENCES Students
      ON DELETE CASCADE
      ON UPDATE SET NULL )
```

General constraints are useful when more general ICs than keys are involved:

```
CREATE TABLE Students
     (sid CHAR(20),
     name CHAR(50),
     email CHAR(30),
     age INTEGER,
     gr INTEGER,
     PRIMARY KEY (sid),
```

```
CONSTRAINT ageInterv
   CHECK (age >= 18
   AND age<=70))
```