Enterprise Computing

Lecture 2: Databases

Agenda

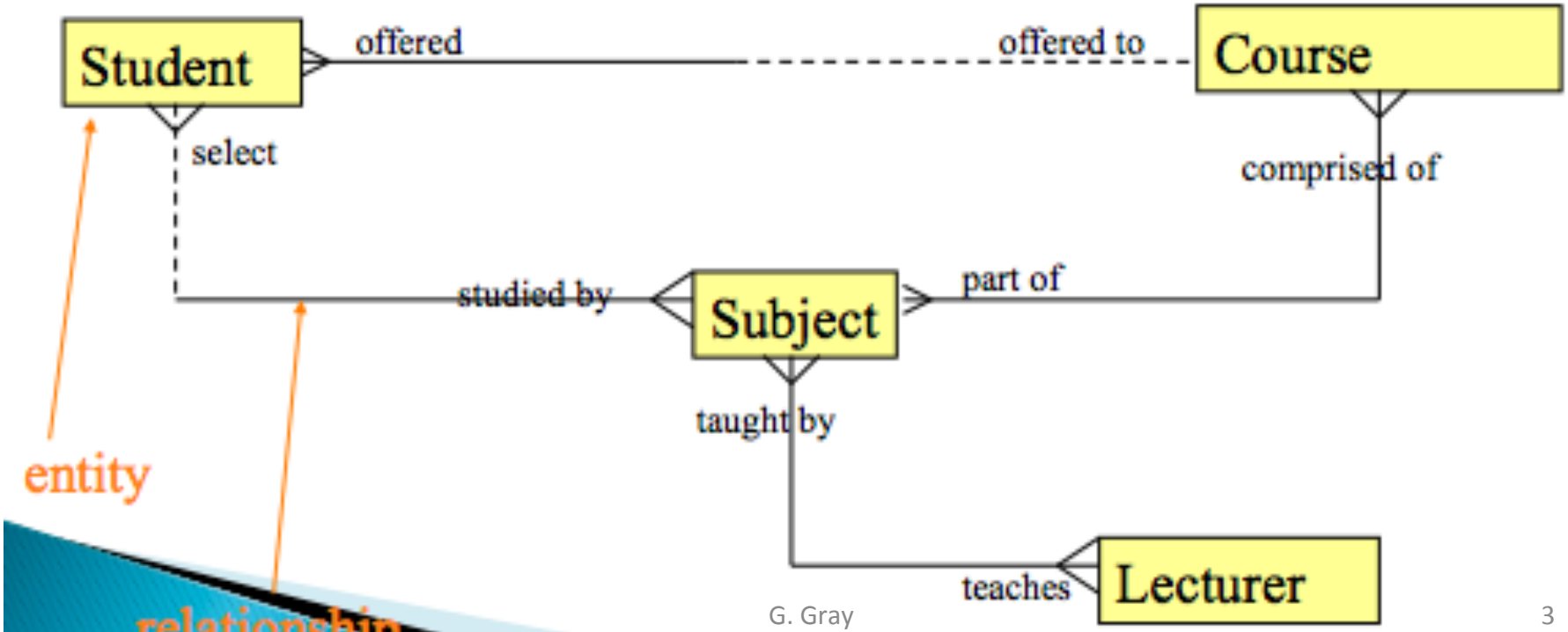
Objective: Revise databases.

Following this lecture you should be able to:

1. Design a database using an ERD
2. Convert your design into SQL create statements to create the database tables.

ERDs

- Entities: nouns in the text identifying what tables are needed in the database
- Relationships: verbs in the text identifying which tables will need to be joined using foreign keys.



Entities

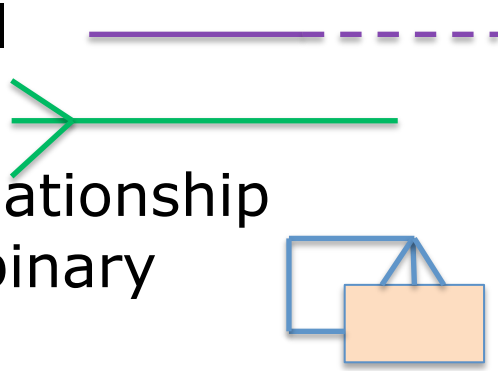
- Entities are the nouns in the text about which you want to store information. They generally fall into four categories:

1. **People**: customer, supplier, employee
2. **Products**: car, book, food item, clothing, etc.
3. **Services**: holiday booking, eating out, hair cut
4. **Recording transactions**: deposit, withdrawal, order, invoice, bill, receipt

Relationships - verbs

- On each relationship, you need to decide:
  - Is the participation mandatory or optional
  - Is the cardinality 1:1, 1:m or m:n

  The degree is the number of entities the relationship joins together, which is usually unary or binary



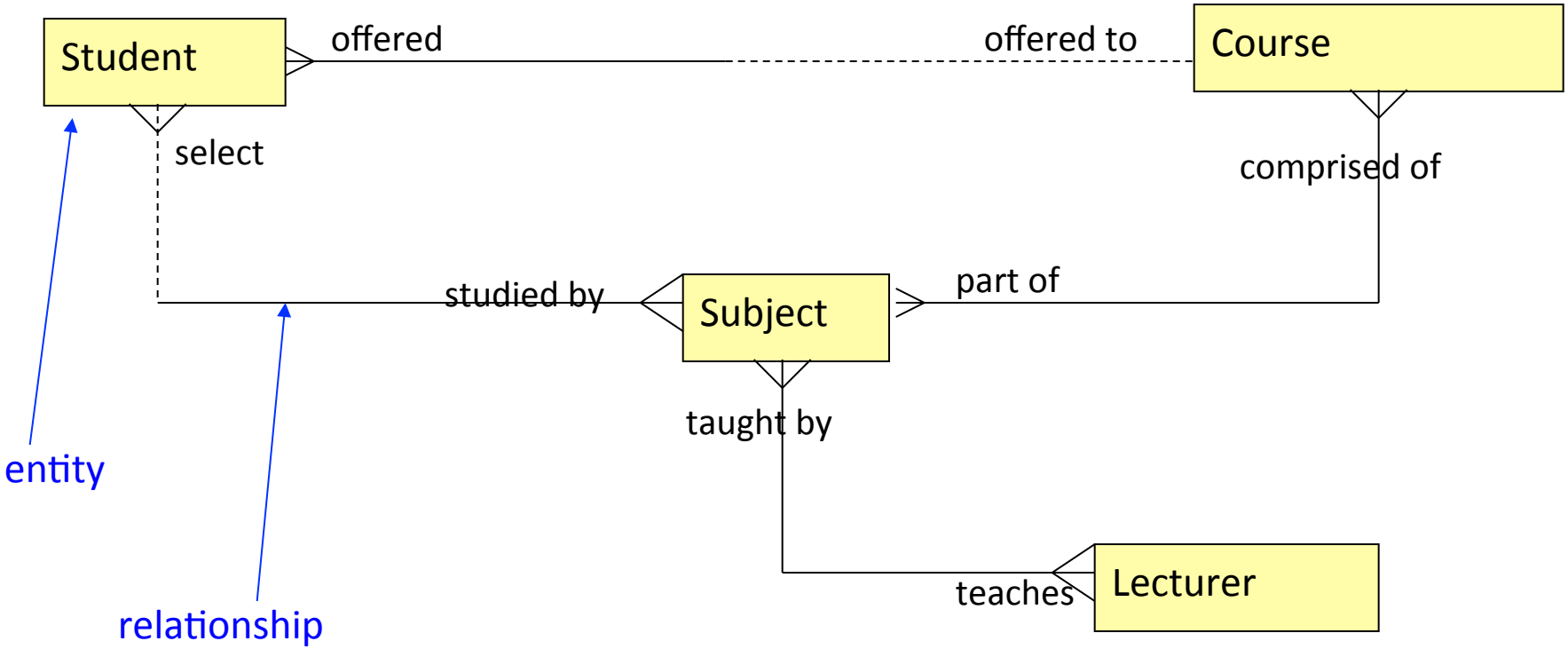
- Attributes: information you want to record about each entity. Attributes can be:
  - Simple or composite
  - Single values or multi-valued
  - Derived
  - The primary key
  - Allowed be NULL

Attributes

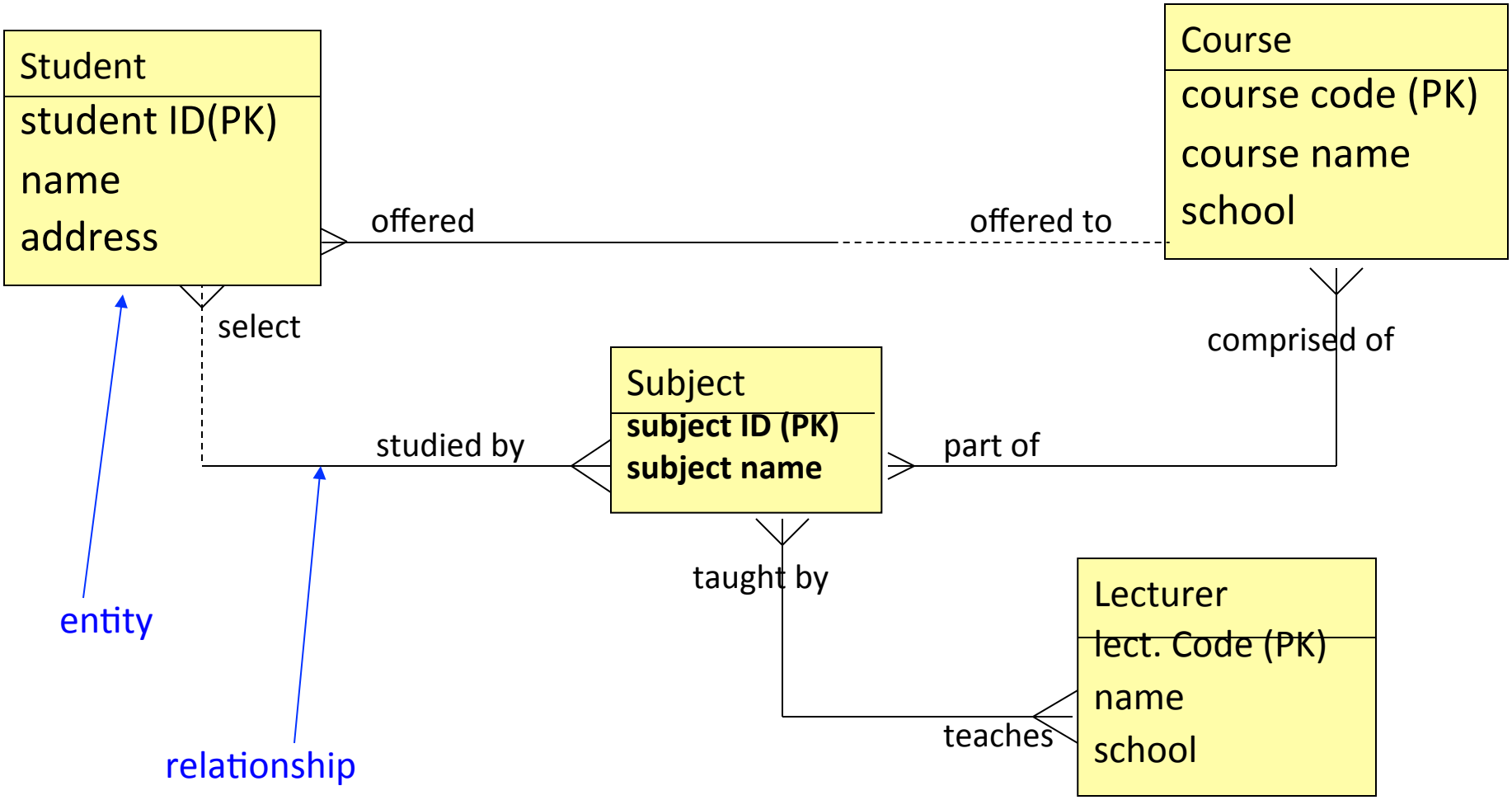
- Attributes are the information you want to record about each entity:
  - People, e.g. employee (name, address, phone number, dateOfBirth)
  - Products: e.g. spareParts(ID, description, quantityInStock, Price)
  - Service: e.g car service(ID, description, price)
  - Transactions: e.g. quote(ID, date, time, cost, etc)

Step 1 - Create ERD

Students are offered a course which is comprised of a number of subjects, each taught by a lecturer. Students can select the subjects they want to do.



Step 2 - Add attributes to ERD



Converting an ERD to a Relational model

1. Each entity type in an ERD becomes a relation in the relational model.
2. Each attribute in an ERD becomes an attribute in the relational model.
3. Relationships in an ERD are represented as Foreign Keys in the relational model.
   - Primary key of the '1' side becomes a foreign key on the 'm' side.
   - n:m relationships must be represented as a new link entity.

Finishing the relational model . . .

- You would also check at this stage:
  - Does every relation have a primary key?
  - Is every attribute **FUNCTIONALLY DEPENDENT** on the primary key (i.e. in the correct table)
  - Is there any composite attributes?
  - Is there any multi-value attributes?
    - If so, create a new relational for the composite attribute with the same primary key as the original attribute
  - Are there duplicate relations – i.e. do two relations have the same (or similar) attributes and can they be merged?
    - e.g. customer and client, or employee and manager . .

See ‘ERD exercises’

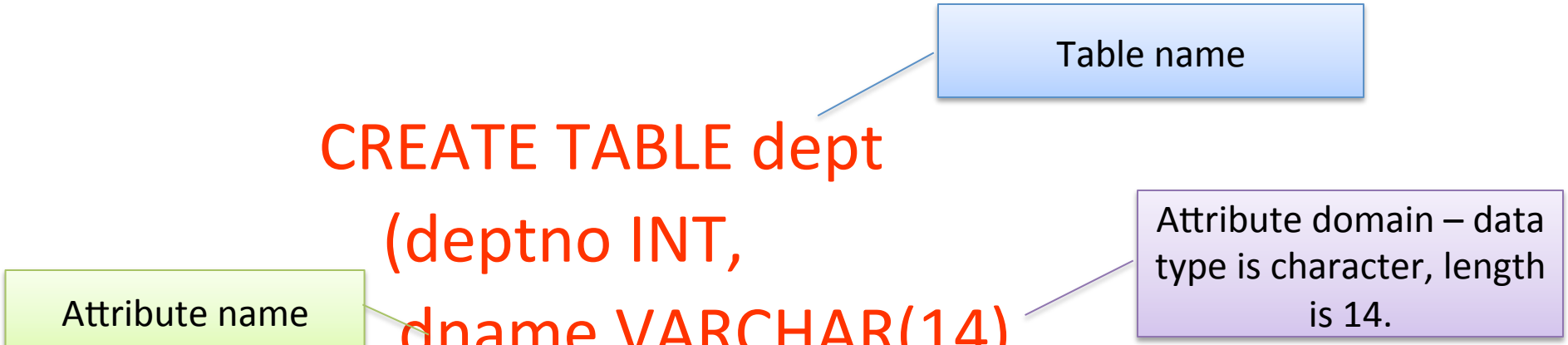
SQL - Table Creation

- To create a table, use the CREATE command which has the following syntax

```
CREATE TABLE table_name
     (column1 datatype [DEFAULT expr] [, . . . ]),
      (column2 datatype,
     ……);
```

- ❖ table is the name of the table
- ❖ column is the name of the attribute
- ❖ datatype is the type and length of the attribute
- ❖ expr specifies a default value for the attribute

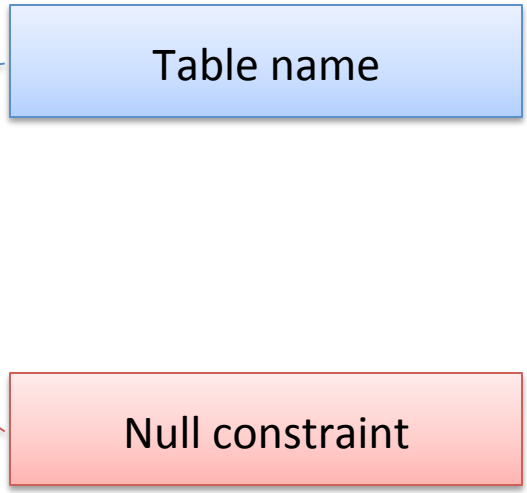
Example of CREATE



Note: SQL statements may end with a semicolon – it's optional in MySQL

Example of CREATE

```
CREATE TABLE emp(
 empno INT NOT NULL,
 ename VARCHAR(10),
 job VARCHAR(9),
 mgr  INT,
 hiredate DATE,
 sal DECIMAL(7,2),
 comm DECIMAL(7,2),
 deptno  INT NOT NULL)
```



Defining the primary key (entity constraint)

Syntax: CONSTRAINT contraint_name PRIMARY KEY (column_name[,column name, . . .])

```
CREATE TABLE DEPT (
 DEPTNO          INT NOT NULL,
 DNAME           VARCHAR(14),
 LOC             VARCHAR(13),
 CONSTRAINT DEPT_PK PRIMARY KEY
   (DEPTNO));
```

Primary key column must be NOT NULL

Each constraint is given a name

The type of constraint, i.e. defining a primary key.

The attribute to be used as the primary key

Defining a foreign key (referential integrity constraint)

```
CREATE TABLE EMP (
 EMPNO              INT NOT NULL,
 ENAME              VARCHAR(10),
 JOB               VARCHAR(9),
 MGR                INT,
 HIREDATE             DATE,
 SAL               DECIMAL(7,2),
 COMM               DECIMAL(7,2),
 DEPTNO             INT NOT NULL,
 CONSTRAINT EMP_DEPTNO_FK FOREIGN KEY
    (DEPTNO) REFERENCES DEPT (DEPTNO),
 CONSTRAINT EMP_MGR_FK FOREIGN KEY
    (MGR) REFERENCES EMP (EMPNO),
 CONSTRAINT EMP_EMPNO_PK PRIMARY KEY (EMPNO));
```

Each constraint is given a name

The type of constraint, i.e. defining a foreign key.

The attribute to be used as the foreign key

The table and attribute the foreign key references

Syntax – foreign key

- The Syntax for defining a foreign key is:

```
CONSTRAINT contraint_name FOREIGN KEY
   (column_name[,column name, . . .])
   REFERENCES table_name
   (primary_key_column[,primary_key_column,
   …]),
```

Primary and Foreign keys – alternative definition

- Primary and Foreign keys can alternatively be declared using inline definitions as follows:

```
CREATE TABLE EMP (
 EMPNO          INT NOT NULL PRIMARY KEY,
 ENAME          VARCHAR(10),
 JOB            VARCHAR(9),
 MGR            INT REFERENCES EMP(EMPNO),
 HIREDAT        DATE,
 SAL            DECIMAL(7,2),
 COM            DECIMAL(7,2),
 DEPTNO         INT NOT NULL REFERENCES
   DEPT(DEPTNO));
```

1. Take a look at ‘Generating SQL from an ERD’ on Moodle.

2. Work on ‘class exercise.doc’