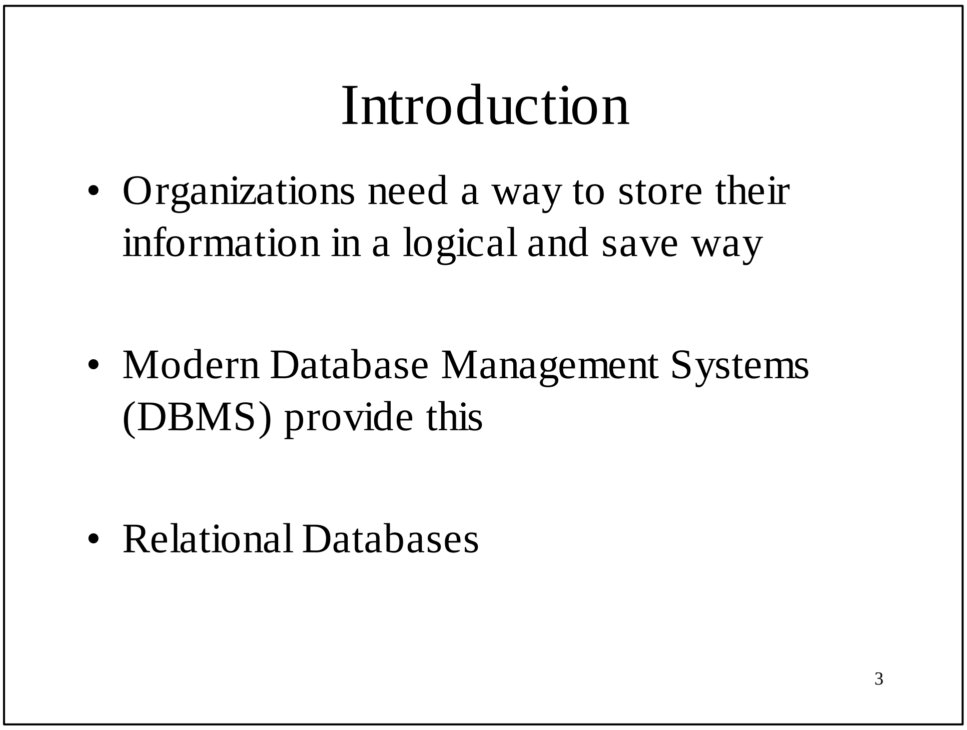
Databases

Michael Hahsler

21.2.2002

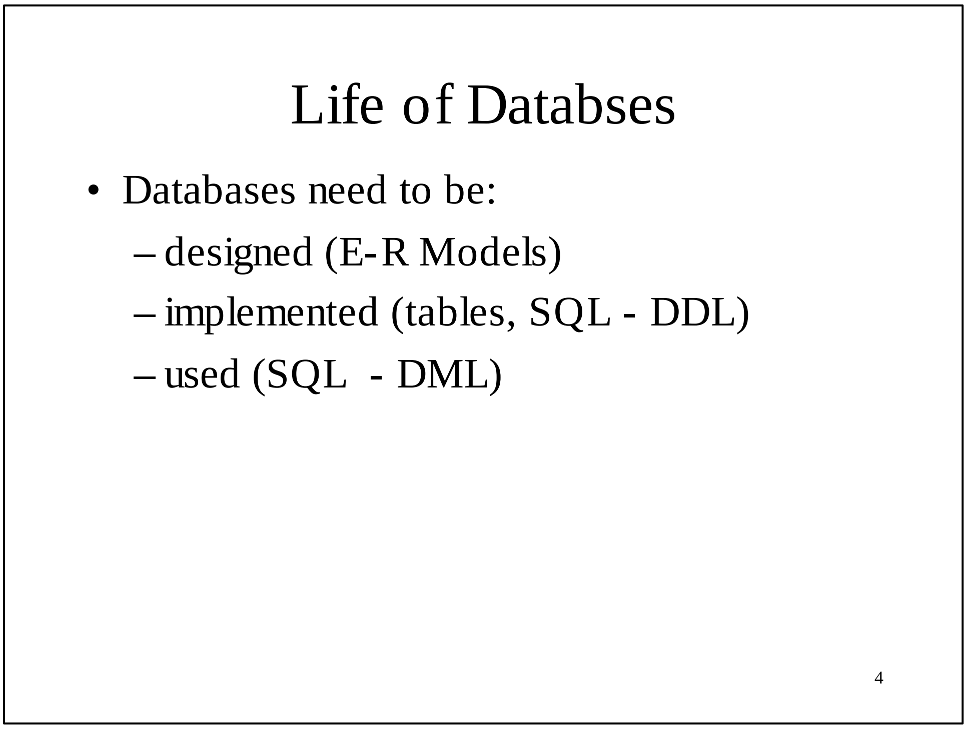
Contents

- DBMS
- Design of Databases
- Entity Relationship Models
- Tables
- SQL



Introduction

- Organizations need a way to store their information in a logical and save way
- Modern Database Management Systems (DBMS) provide this
- Relational Databases



Life of Databses

- Databases need to be:
  - designed (E-R Models)
  - implemented (tables, SQL - DDL)
  - used (SQL - DML)

Design of a Grades DB

- I am a teacher and want to keep track of the grades of my students in a database
- I teach several classes
- Students can take several classes with me

A simple Table

Grades

Class	Student	Grade
COAP 2120	Peter	B
COAP 2120	Monica	A
COAP 9000	Peter	F

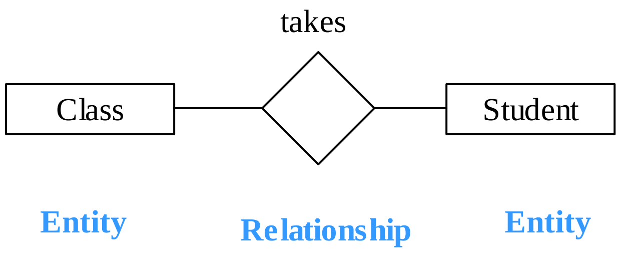
Design of a Grades DB II

- I am a **teacher** and *want* to keep track of the **grades** of my **students** in a **database**
- I *teach* several **classes**
- **Students** can *take* several **classes** with me

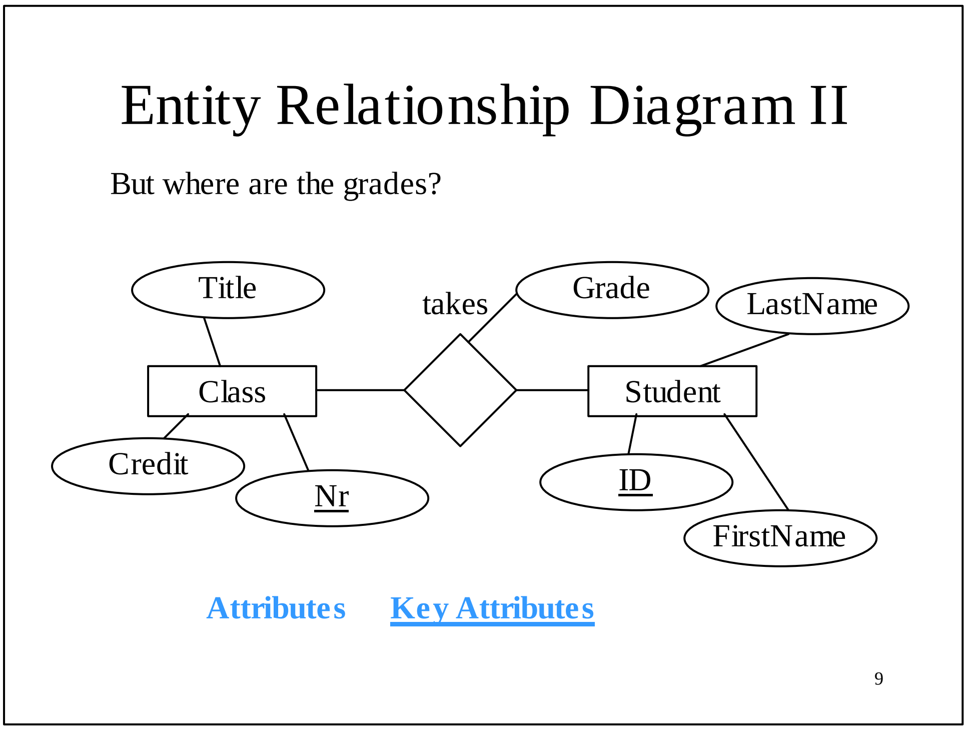
nouns - Entities/Objects

verbs - Relationships

Entity Relationship Diagram I



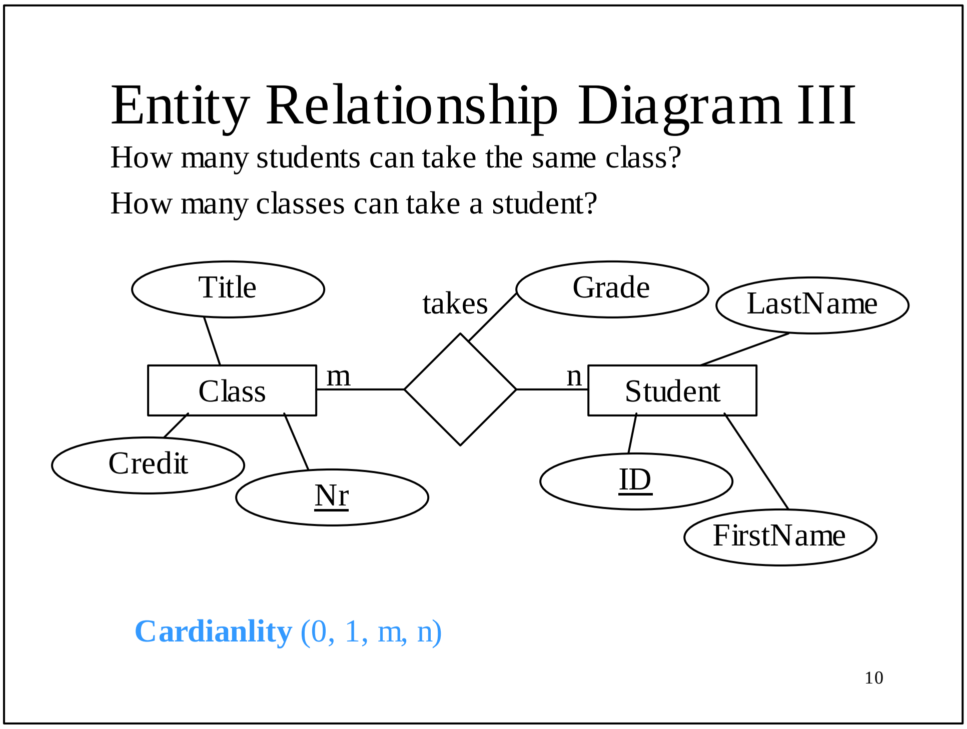
But where are the grades?



Entity Relationship Diagram II

But where are the grades?

Attributes **Key Attributes**



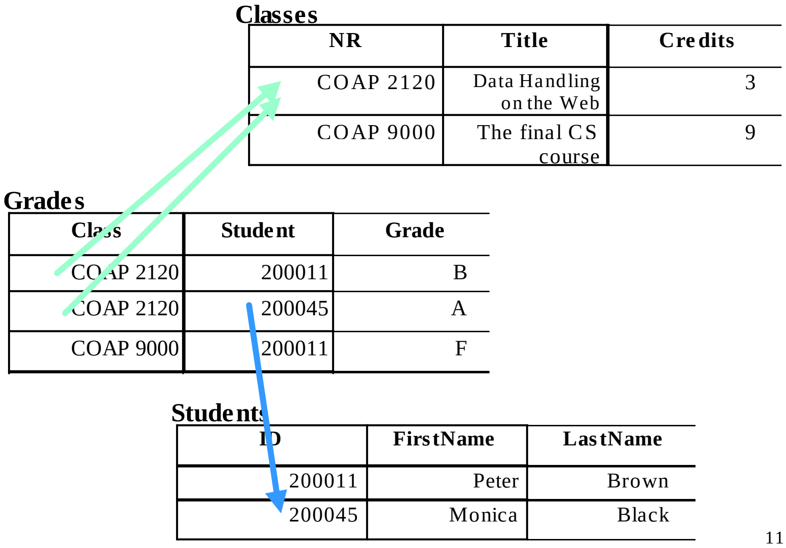
Entity Relationship Diagram III

How many students can take the same class?

How many classes can take a student?

Cardianlity (0, 1, m, n)

Result: Serveral Tables



Classes

NR	Title	Credits
COAP 2120	Data Handling on the Web	3
COAP 9000	The final CS course	9

Grades

Class	Student	Grade
COAP 2120	200011	B
COAP 2120	200045	A
COAP 9000	200011	F

Students

ID	FirstName	LastName
200011	Peter	Brown
200045	Monica	Black

Example

- You work for a Internet company that does Web-site development. For each project your company charges the client the needed hours. You need a DB to track the hours spent on each project.
- Each developer can work on several projects. On each project several developer can work. Each project has one leader who is one of the developers

Implementration: SQL

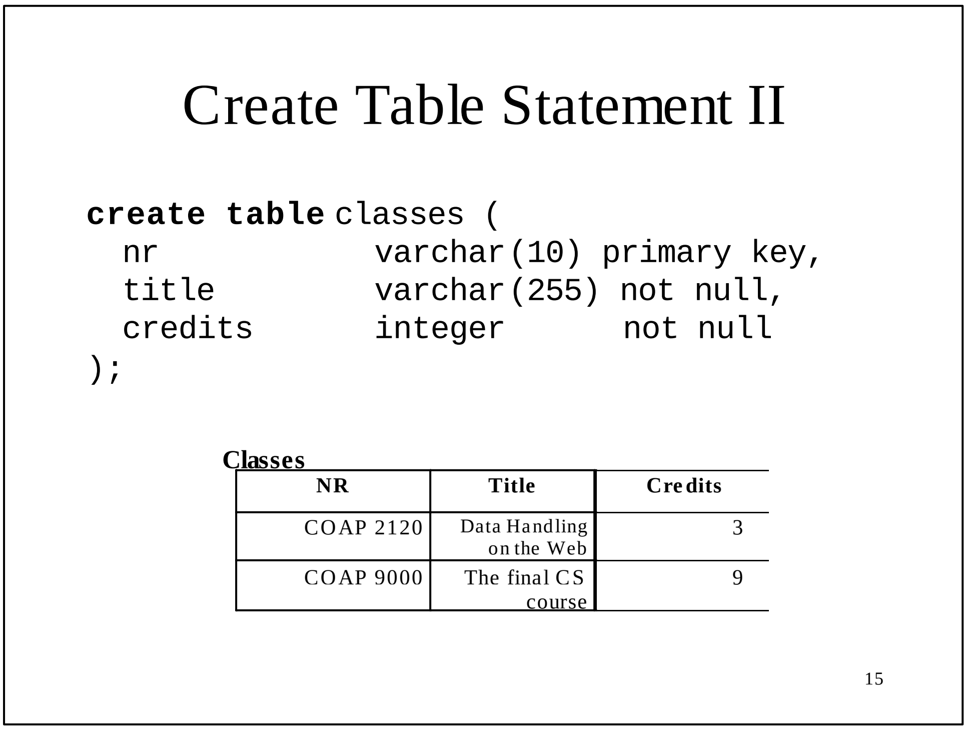
- Structured Query Language consists of
  - Data Definition Language (DDL)
    Define tables with attributes in your DB (`create`)
  - Data Manipultation Language (DML)
    Enter data into your DB and get data out of your DB (`insert, select`)

Create Table Statement I

```
create table students (
  id            integer   primary key,
  firstname     varchar(255),
  lastname      varchar(255) not null
);
```

Students

ID	FirstName	LastName
200011	Peter	Brown
200045	Monica	Black

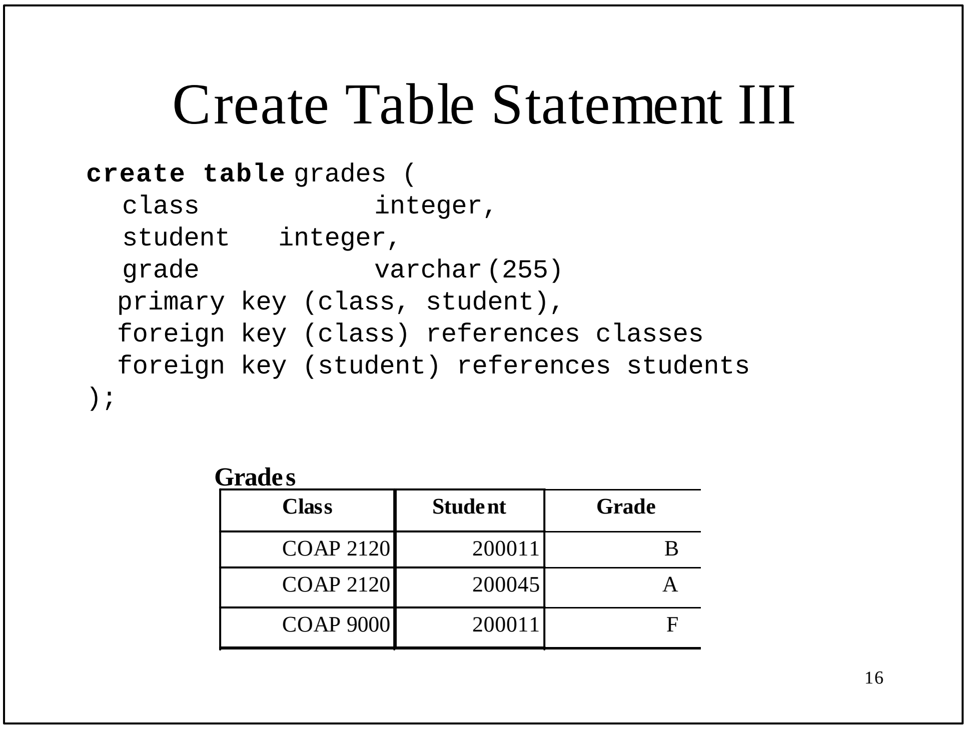


Create Table Statement II

```
create table classes (
  nr            varchar(10) primary key,
  title         varchar(255) not null,
  credits       integer      not null
);
```

Classes

NR	Title	Credits
COAP 2120	Data Handling on the Web	3
COAP 9000	The final CS course	9

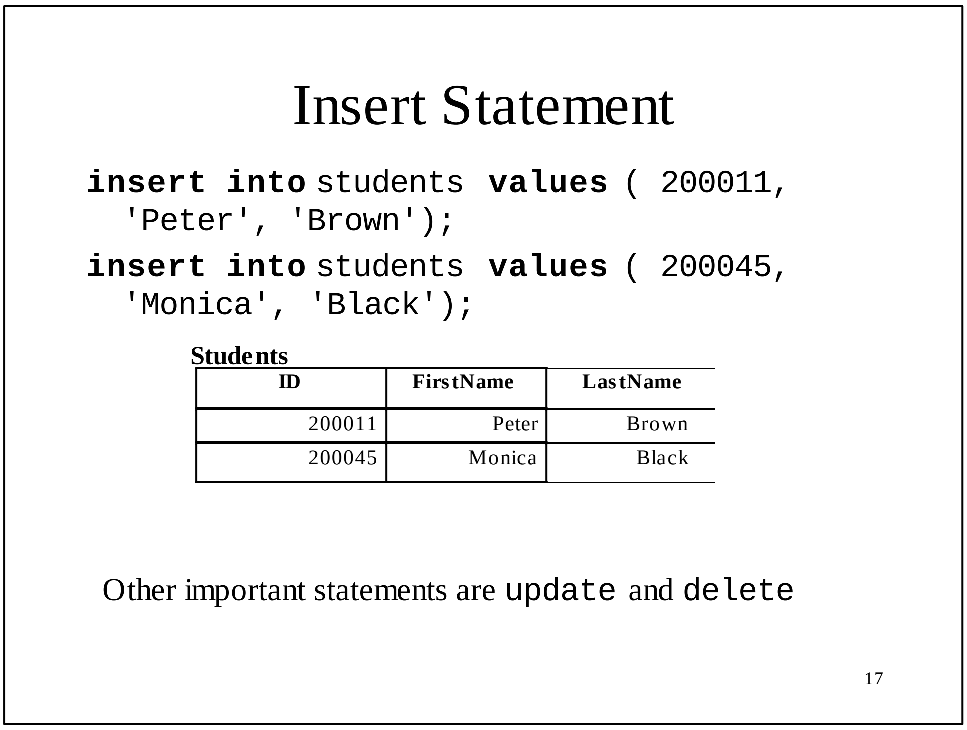


Create Table Statement III

```
create table grades (
  class           integer,
  student    integer,
  grade           varchar(255)
  primary key (class, student),
  foreign key (class) references classes
  foreign key (student) references students
);
```

Grades

Class	Student	Grade
COAP 2120	200011	B
COAP 2120	200045	A
COAP 9000	200011	F



Insert Statement

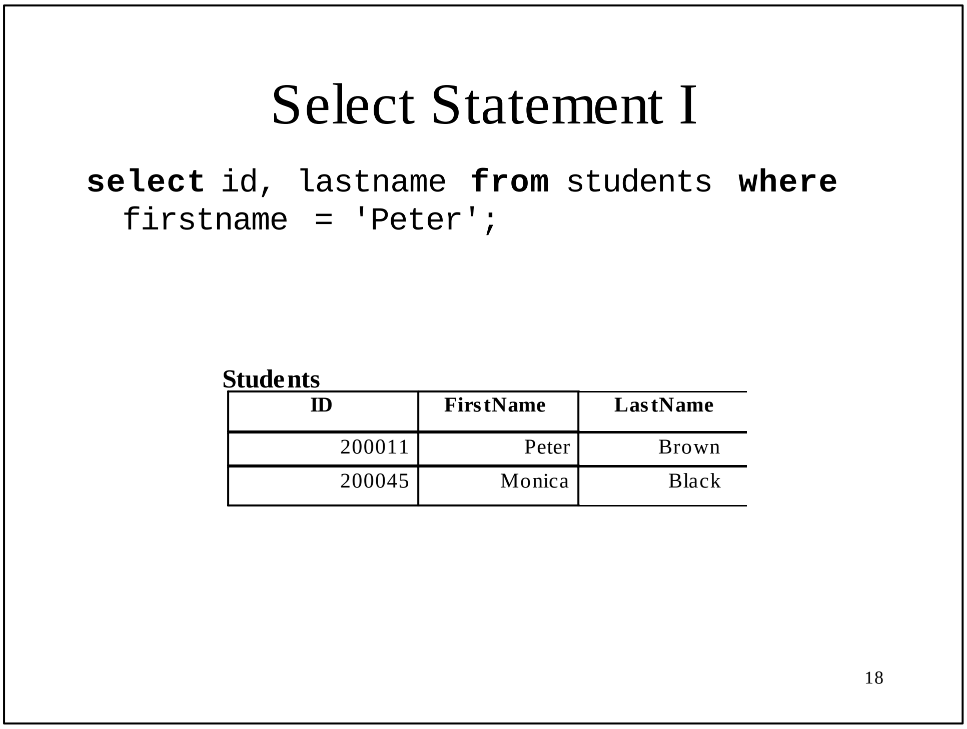
```
insert into students values ( 200011,
  'Peter', 'Brown');
insert into students values ( 200045,
  'Monica', 'Black');
```

Students

ID	FirstName	LastName
200011	Peter	Brown
200045	Monica	Black

Other important statements are `update` and `delete`

17



Select Statement I

```
select id, lastname from students where
  firstname = 'Peter';
```

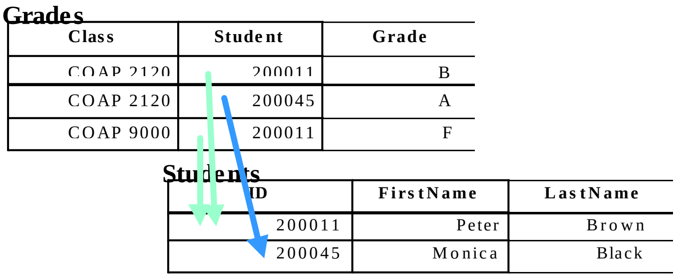
Students

ID	FirstName	LastName
200011	Peter	Brown
200045	Monica	Black

18

Select Statement II - join

```
select lastname, class from grades,
  students where
  grades.student=students.idand
  grade='A' or grade='B' order by grade;
```



Grades

Class	Student	Grade
COAP 2120	200011	B
COAP 2120	200045	A
COAP 9000	200011	F

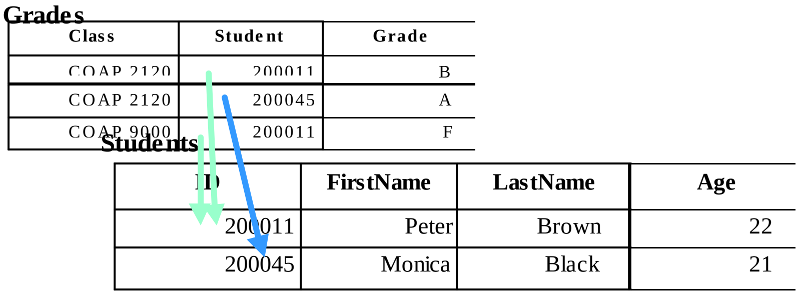
Students

ID	FirstName	LastName
200011	Peter	Brown
200045	Monica	Black

19

Select Statement III - groups

```
select grade, avg(age) from grades,
  students where student=id group by
  grade order by grade;
```



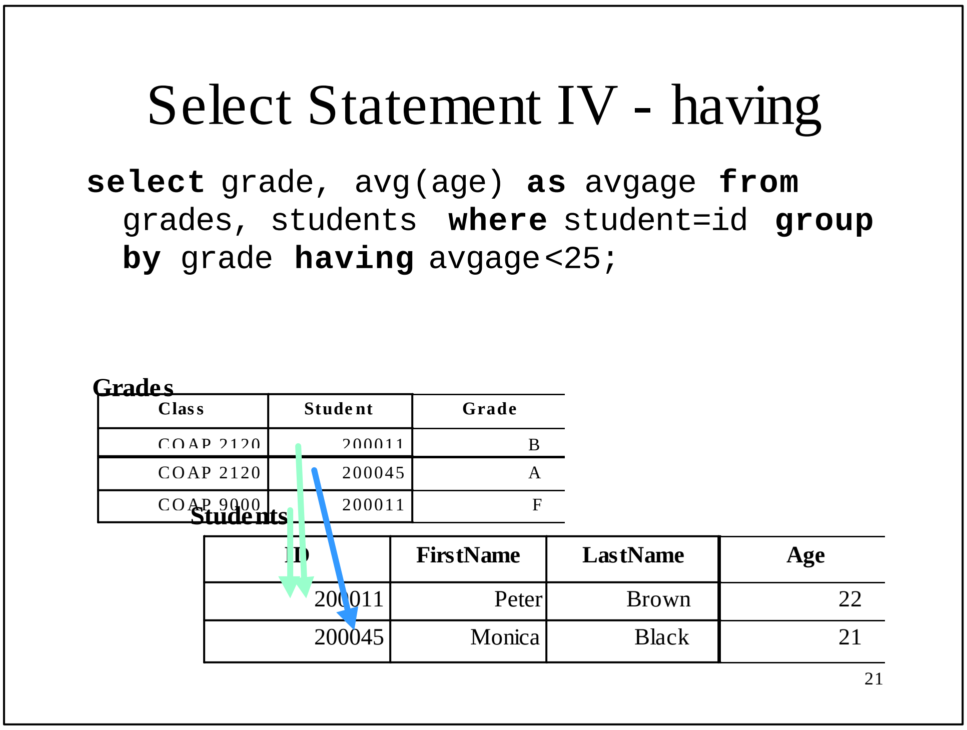
Grades

Class	Student	Grade
COAP 2120	200011	B
COAP 2120	200045	A
COAP 9000	200011	F

Students

ID	FirstName	LastName	Age
200011	Peter	Brown	22
200045	Monica	Black	21

20



Select Statement IV - having

```
select grade, avg(age) as avgage from
  grades, students where student=id group
  by grade having avgage<25;
```

Grades

Class	Student	Grade
COAP 2120	200011	B
COAP 2120	200045	A
COAP 9000	200011	F

Students

ID	FirstName	LastName	Age
200011	Peter	Brown	22
200045	Monica	Black	21

21