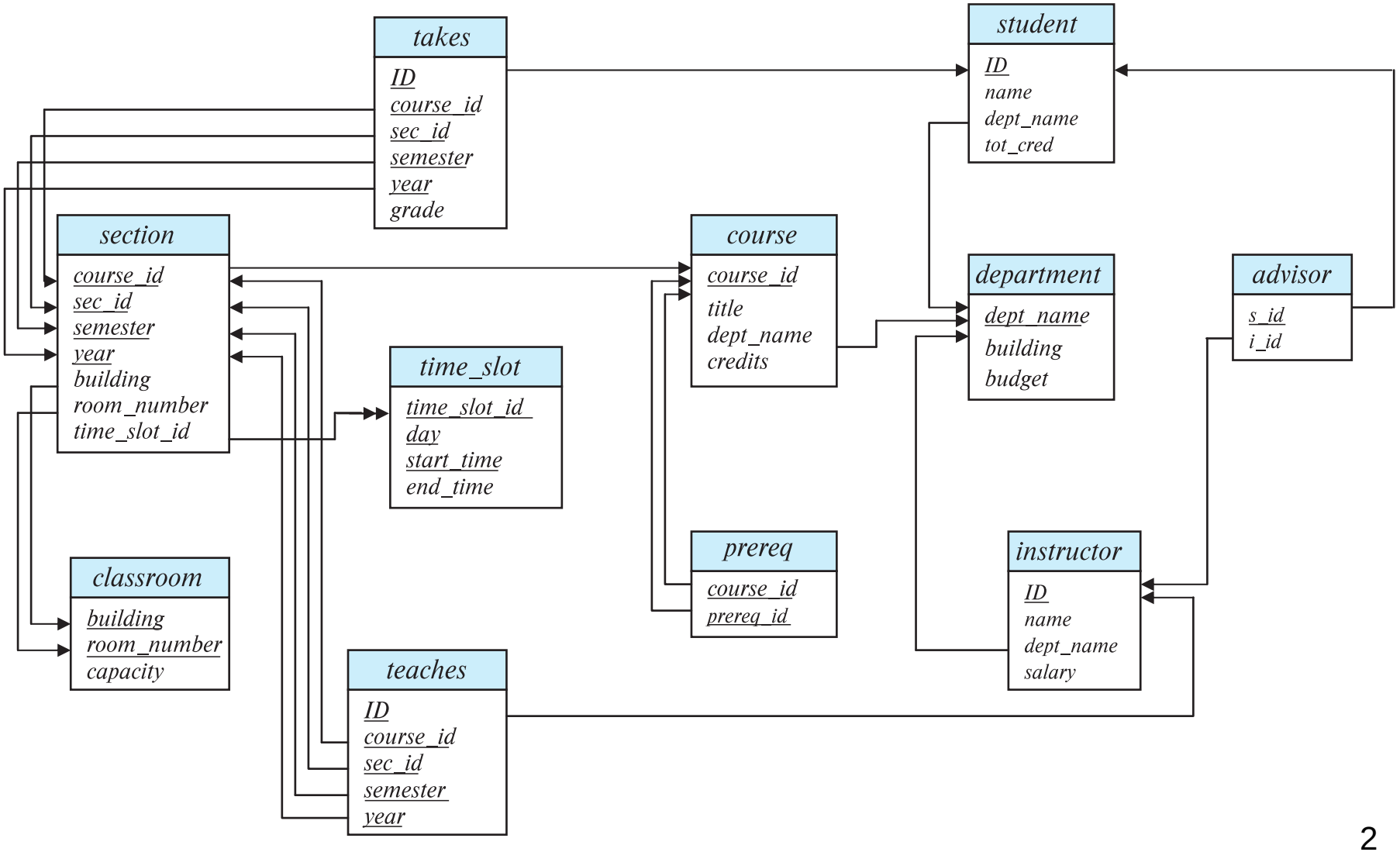
Structured Query Language (SQL)

Miao Qiao
The University of Auckland

Recap: Schema for University Database



Recap: Relational Algebra

- A query language of a set of operations that take one or two relations as input and produce a new relation as their result.
- Operators
  - select: σ
  - project: Π
  - union: $\cup$
  - set difference: –
  - Cartesian product: x
    - Natural Join $\bowtie$
    - Theta join $\bowtie_\theta$
  - rename: ρ
  - assignment : $\leftarrow$
- Expression tree

History: Structured Query Language (SQL)

- ANSI and ISO standard SQL:
  - SQL-86
  - SQL-89
  - SQL-92
  - SQL:1999 (language name became Y2K compliant!)
  - SQL:2003

Data Definition Language (DDL)

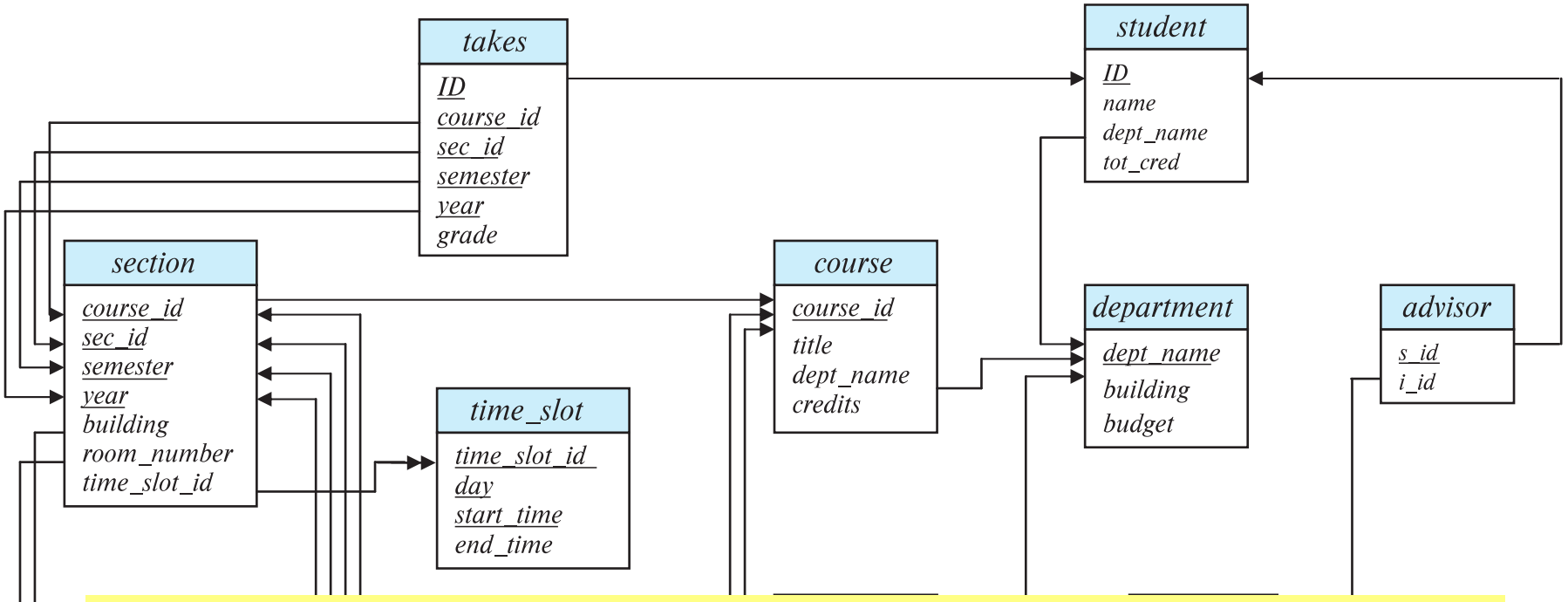
- Define database schema

Example: **create table** *instructor* (
ID **char**(5),
name **varchar**(20),
dept_name **varchar**(20),
salary **numeric**(8,2))

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

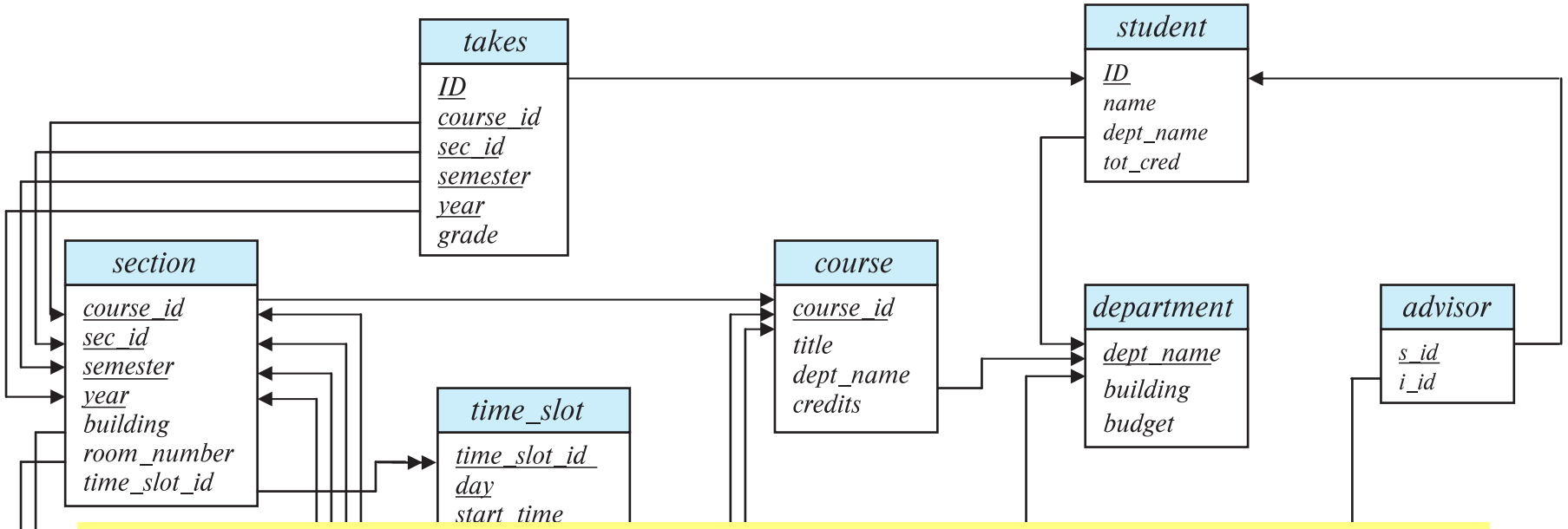
(a) The *instructor* table

University Database: Create table for student



- **create table** *student* (
  ID **varchar**(5),
  name **varchar**(20) not null,
  dept_name **varchar**(20),
  tot_cred **numeric**(3,0),
  primary key *(ID)*,
  foreign key *(dept_name)* **references** *department*);

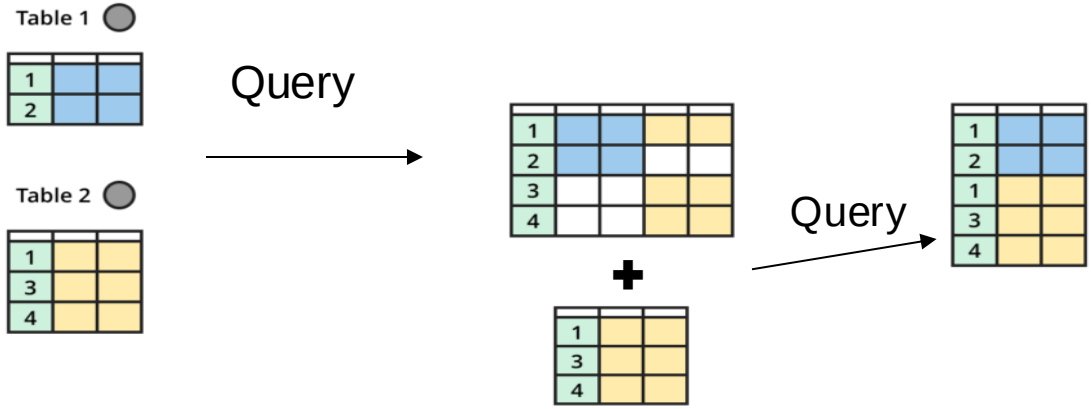
University Database: Create table for takes



- **create table** *takes* (
  ID **varchar**(5),
  course_id **varchar**(8),
  sec_id **varchar**(8),
  semester **varchar**(6),
  year **numeric**(4,0),
  grade **varchar**(2),
  primary key *(ID, course_id, sec_id, semester, year)* ,
  foreign key (*ID*) **references** *student*,
  foreign key (*course_id, sec_id, semester, year*) **references** *section*);

Data Manipulation Language (DML)

- Language for accessing and updating the data
- DML also known as **query language**
- **Declarative**
- Query returns a relation => compositional and close
- Compositional



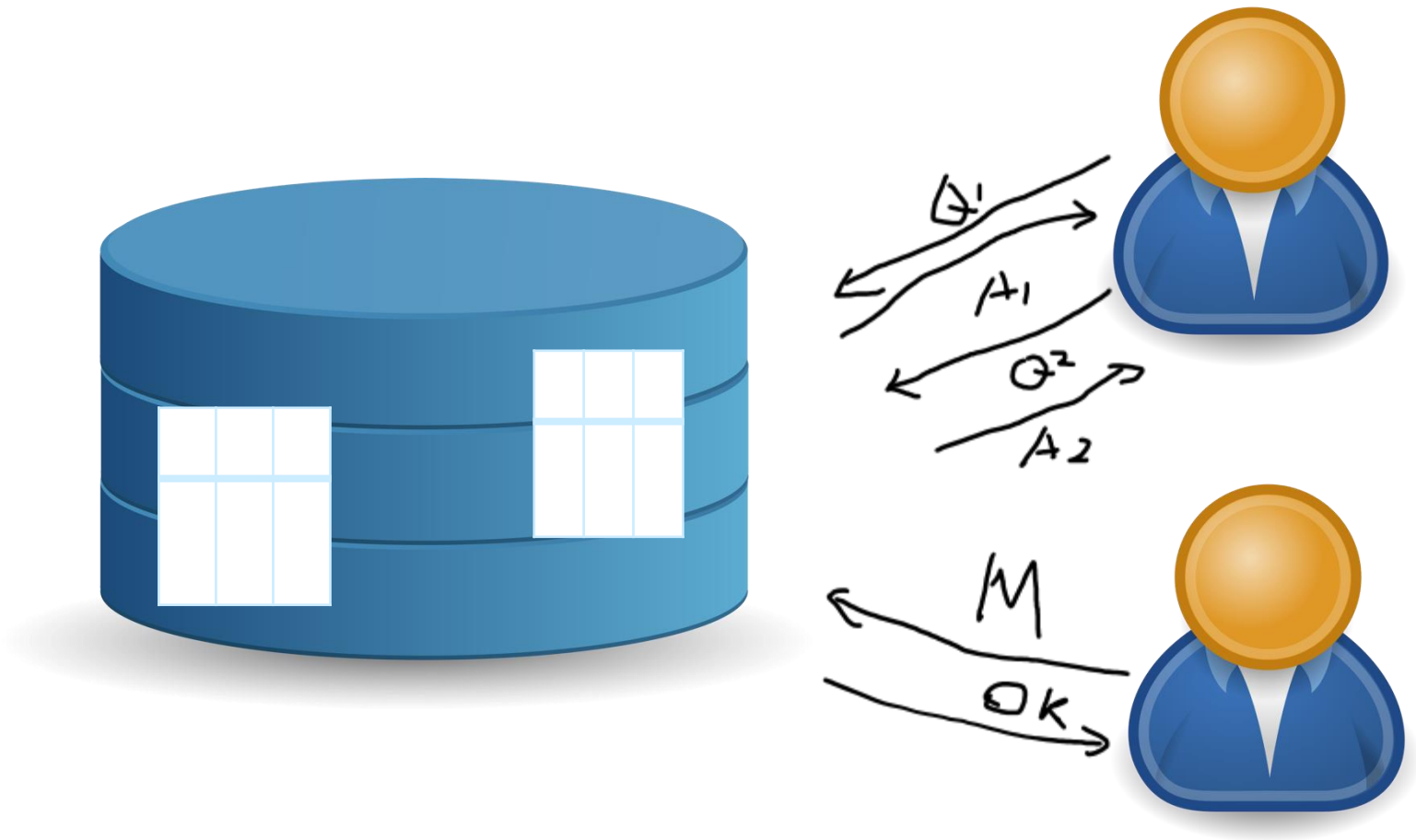
Basic Query Structure

- A typical SQL query has the form:

select A_1, A_2, ..., A_n
from r_1, r_2, ..., r_m
where P

  - A_i represents an attribute
  - r_i represents a relation
  - P is a predicate.
- The result of an SQL query is a relation.
- Update
  - **insert into** *instructor* **values** ('10211', 'Smith', 'Biology', 66000);
  - **delete from** *student*
  - **drop table** *student*
  - alter
    - **alter table** *student* **add** *age numeric(0,150)*
    - **alter table** *student* **drop** *age*

Basic steps in creating and using relation DB



Demonstrations

- Basic Query Structure of SQL Queries
- Additional Basic Operations
- Set Operations
- Null Values
- Aggregate Functions
- Nested Subqueries
- Modification of the Database

Demonstrations

- Join Expressions
- Views
- Transactions
- Integrity Constraints
- SQL Data Types and Schemas
- Index Definition in SQL
- Authorization

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

SCIENCE

FIN

Any questions?