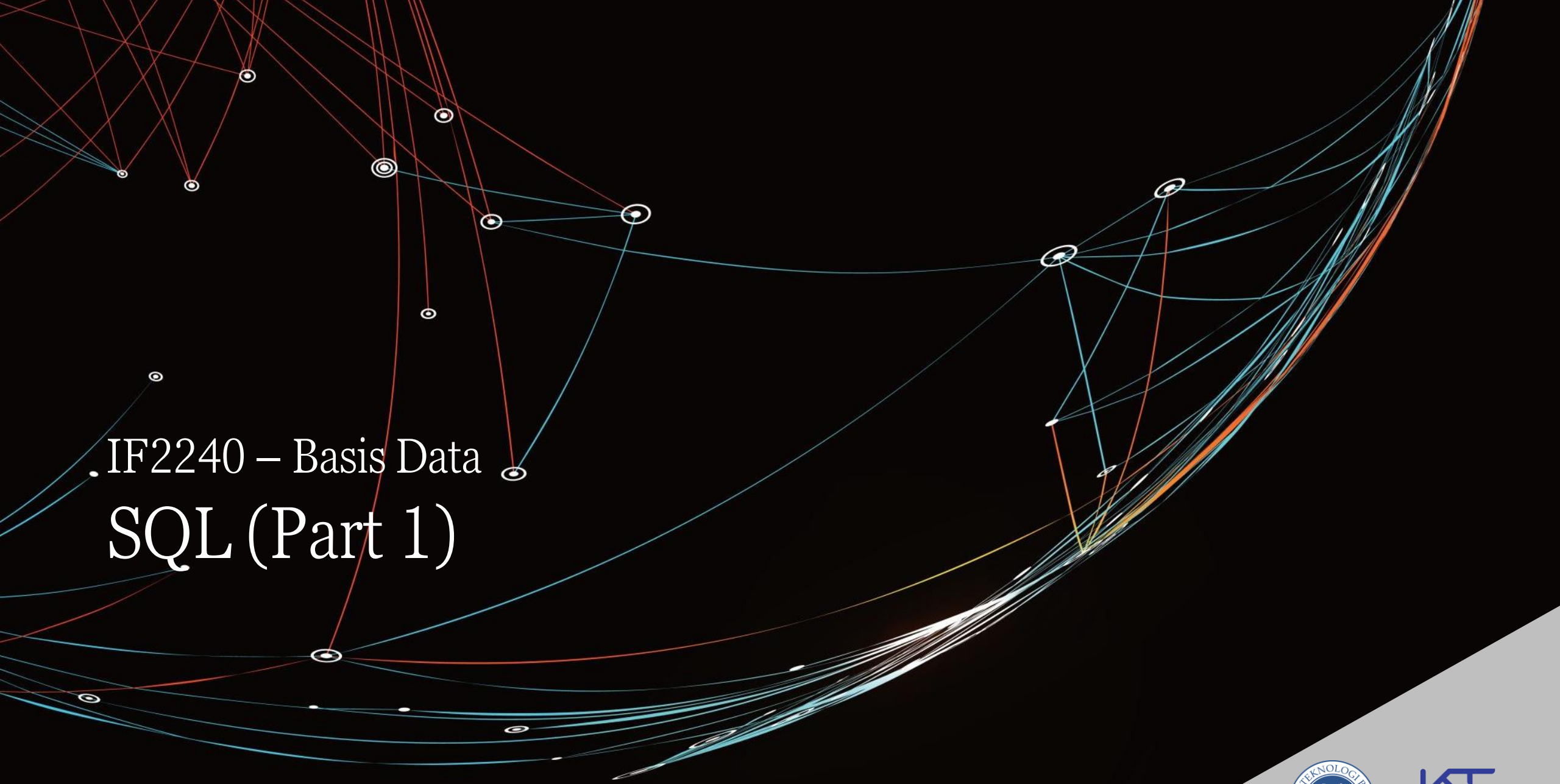
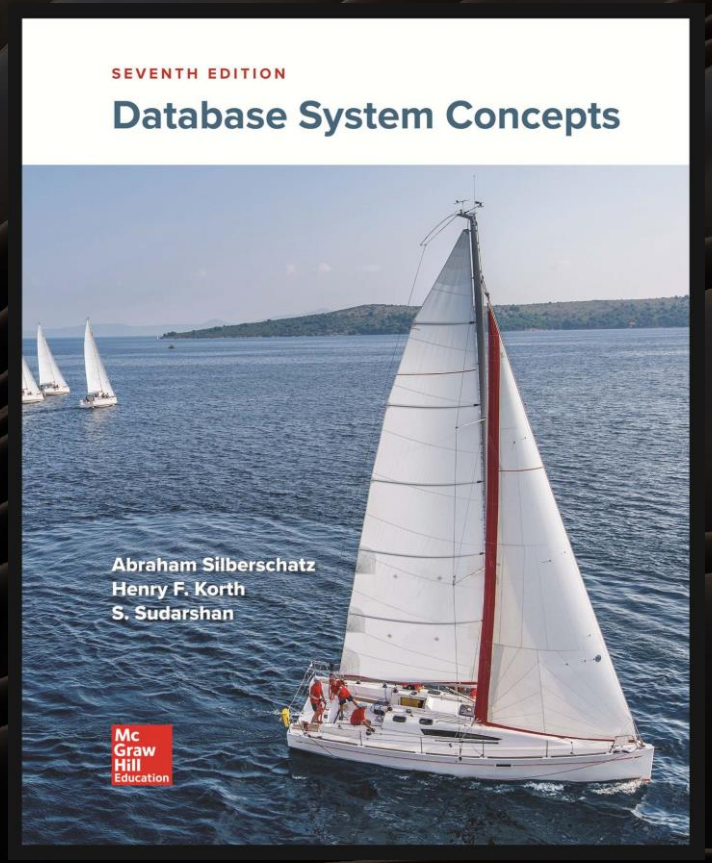


IF2240 – Basis Data

SQL (Part 1)

©2023 - Tim Pengajar IF2240 Semester I TA 2022/2023

KNOWLEDGE & SOFTWARE ENGINEERING



Sumber

Silberschatz, Korth, Sudarshan: "Database System Concepts", 7th Edition

- Chapter 3 : Introduction to SQL

©2022 - Tim Pengajar IF2240 Basis Data

History

IBM Sequel language developed as part of System R project at the IBM San Jose Research Laboratory

Renamed Structured Query Language (SQL)

ANSI and ISO standard SQL:

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

Commercial systems offer most, if not all, SQL-92 features, plus varying feature sets from later standards and special proprietary features.

- Not all examples here may work on your particular system.

SQL Parts

DML -- provides the ability to query information from the database and to insert tuples into, delete tuples from, and modify tuples in the database.

integrity – the DDL includes commands for specifying integrity constraints.

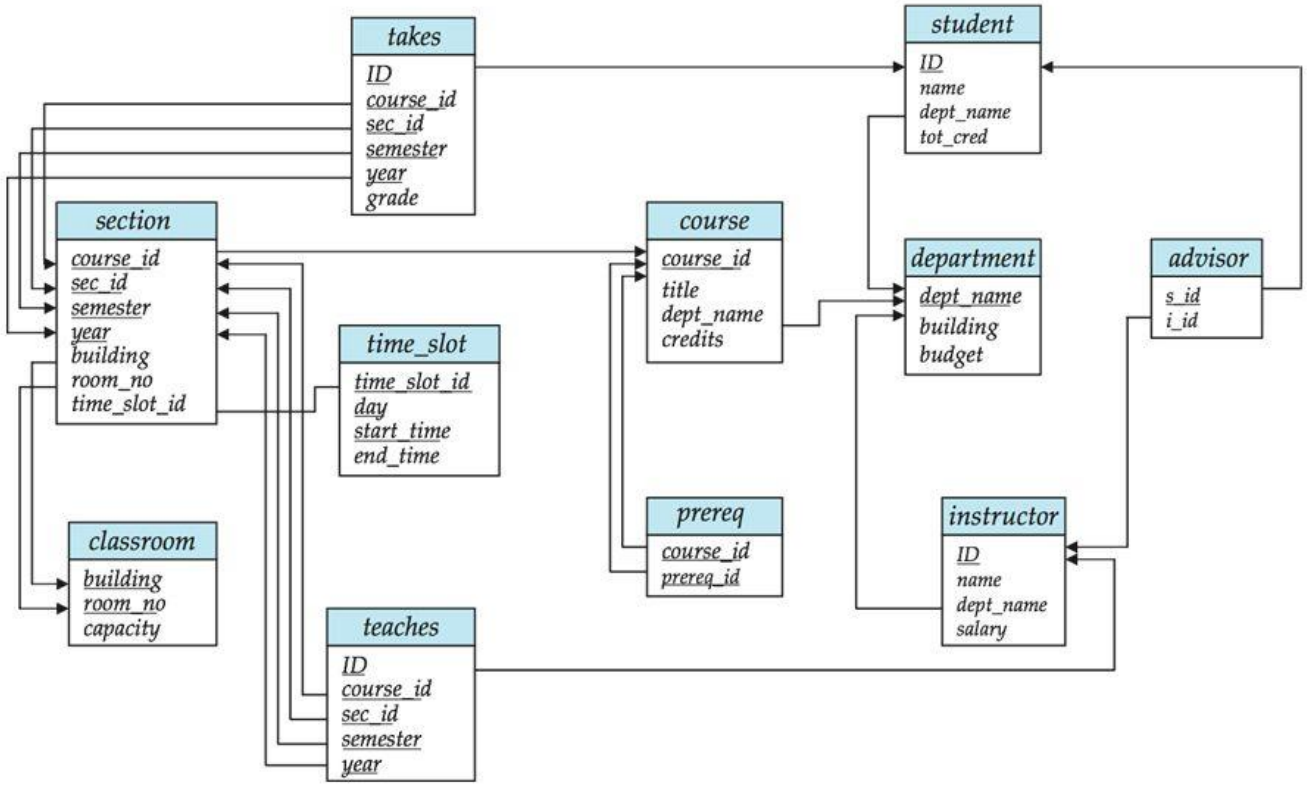
View definition -- The DDL includes commands for defining views.

Transaction control –includes commands for specifying the beginning and ending of transactions.

Embedded SQL and dynamic SQL -- define how SQL statements can be embedded within general-purpose programming languages.

Authorization – includes commands for specifying access rights to relations and views.

Schema Diagram



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.

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

The select Clause

The **select** clause lists the attributes desired in the result of a query

- corresponds to the projection operation of the relational algebra

Example: find the names of all instructors:

select *name*
from *instructor*

* utk smua atribut (select *)

NOTE: SQL names are case insensitive (i.e., you may use upper- or lower-case letters.)

- E.g., *Name* ≡ *NAME* ≡ *name*
- Some people use upper case wherever we use bold font.

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

The select Clause (Cont.)

SQL allows duplicates in relations as well as in query results.

To force the elimination of duplicates, insert the keyword **distinct** after select.

Find the department names of all instructors, and remove duplicates

```
select distinct dept_name
from instructor
```

The keyword **all** specifies that duplicates should not be removed.

```
select all dept_name
from instructor
```

dept_name
Comp. Sci.
Finance
Music
Physics
History
Physics
Comp. Sci.
History
Finance
Biology
Comp. Sci.
Elec. Eng.

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

The select Clause (Cont.)

An asterisk in the select clause denotes "all attributes"

```
select *
from instructor
```

* artinya select semua

An attribute can be a literal with no **from** clause : **select** '437'

- Results is a table with one column and a single row with value "437"
- Can give the column a name using:

```
select '437' as FOO
```

An attribute can be a literal with **from** clause

```
select 'A'
from instructor
```

- Result is a table with one column and *N* rows (number of tuples in the *instructors* table), each row with value "A"

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

The select Clause (Cont.)

The **select** clause can contain arithmetic expressions involving the operation, +, –, *, and /, and operating on constants or attributes of tuples.

- The query:

```
select ID, name, salary/12
 from instructor
```

  would return a relation that is the same as the *instructor* relation, except that the value of the attribute *salary* is divided by 12.

- Can rename "*salary/12*" using the **as** clause:

```
select ID, name, salary/12  as monthly_salary
```


©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

The where Clause

The **where** clause specifies conditions that the result must satisfy

- Corresponds to the selection predicate of the relational algebra.

To find all instructors in Comp. Sci. dept

```
select name
from instructor
where dept_name = 'Comp. Sci.'
```

SQL allows the use of the logical connectives **and, or,** and **not.** The operands of the logical connectives can be expressions involving the comparison operators <, <=, >, >=, =, and <>. Comparisons can be applied to results of arithmetic expressions

To find all instructors in Comp. Sci. dept with salary > 70000

```
select name
from instructor
where dept_name = 'Comp. Sci.' and salary > 70000
```

name
Katz
Brandt

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

The from Clause

The **from** clause lists the relations involved in the query

- Corresponds to the Cartesian product operation of the relational algebra.

Find the Cartesian product *instructor X teaches*

```
select *
from instructor, teaches
```

- generates every possible instructor – teaches pair, with all attributes from both relations.
- For common attributes (e.g., *ID*), the attributes in the resulting table are renamed using the relation name (e.g., *instructor.ID*)

Cartesian product not very useful directly, but useful combined with where-clause condition (selection operation in relational algebra).

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

Examples

Find the names of all instructors who have taught some course and the course_id

- **select** *name, course_id*
  from *instructor , teaches*
  where *instructor.ID = teaches.ID*

Find the names of all instructors in the Art department who have taught some course and the course_id

- **select** *name, course_id*
  from *instructor , teaches*
  where *instructor.ID = teaches.ID*
  and *instructor. dept_name* = 'Art'

name	*course_id*
Srinivasan	CS-101
Srinivasan	CS-315
Srinivasan	CS-347
Wu	FIN-201
Mozart	MU-199
Einstein	PHY-101
El Said	HIS-351
Katz	CS-101
Katz	CS-319
Crick	BIO-101
Crick	BIO-301
Brandt	CS-190
Brandt	CS-190
Brandt	CS-319
Kim	EE-181

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

The Rename Operation

The SQL allows renaming relations and attributes using the **as** clause:

old-name **as** *new-name*

Find the names of all instructors who have a higher salary than some instructor in 'Comp. Sci'.

- **select distinct** *T.name*
  from *instructor* **as** *T, instructor* **as** *S*
  where *T.salary* > *S.salary* **and** *S.dept_name* = *'Comp. Sci.'*

Keyword **as** is optional and may be omitted

instructor **as** *T* ≡ *instructor T*

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

Self "Join" Example

Relation *emp-super*

person	*supervisor*
Bob	Alice
Mary	Susan
Alice	David
David	Mary

Find the supervisor of "Bob"

Find the supervisor of the supervisor of "Bob"

Can you find ALL the supervisors (direct and indirect) of "Bob"?

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

String Operations

SQL includes a string-matching operator for comparisons on character strings. The operator **like** uses patterns that are described using two special characters:

- percent (%). The % character matches any substring.
- underscore (_). The _ character matches any character.

Find the names of all instructors whose name includes the substring "dar".

```
select name
from instructor
where name like '%dar%'
```

Match the string "100%"

```
like '100 \%'  escape  '\'
```

in that above we use backslash (\) as the escape character.

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

String Operations (Cont.)

Patterns are case sensitive.

Pattern matching examples:

- 'Intro%' matches any string beginning with "Intro".
- '%Comp%' matches any string containing "Comp" as a substring.
- '_ _ _' matches any string of exactly three characters.
- '_ _ _ %' matches any string of at least three characters.

SQL supports a variety of string operations such as

- concatenation (using "||")
- converting from upper to lower case (and vice versa)
- finding string length, extracting substrings, etc.

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

Ordering the Display of Tuples

List in alphabetic order the names of all instructors

```
select distinct name
from    instructor
order by name
```

We may specify **desc** for descending order or **asc** for ascending order, for each attribute; ascending order is the default.

- Example: **order by** *name* **desc**

Can sort on multiple attributes

- Example: **order by** *dept_name, name*

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

Where Clause Predicates

SQL includes a **between** comparison operator

Example: Find the names of all instructors with salary between $90,000 and $100,000 (that is, ≥ $90,000 and ≤ $100,000)

- **select** *name*
  from *instructor*
  where *salary* **between** 90000 **and** 100000

Tuple comparison

- **select** *name*, *course_id*
  from *instructor*, *teaches*
  where (*instructor.ID*, *dept_name*) = (*teaches.ID*, 'Biology');

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

Set Operations

- Find courses that ran in Fall 2017 **or** in Spring 2018

```
(select course_id  from section where sem = 'Fall' and year = 2017)
union
 (select course_id  from section where sem = 'Spring' and year = 2018)
```

- Find courses that ran in Fall 2017 **and** in Spring 2018

```
(select course_id  from section where sem = 'Fall' and year = 2017)
intersect
 (select course_id  from section where sem = 'Spring' and year = 2018)
```

- Find courses that ran in Fall 2017 **but not in** Spring 2018

```
(select course_id  from section where sem = 'Fall' and year = 2017)
except
 (select course_id  from section where sem = 'Spring' and year = 2018)
```


©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

Set Operations (Cont.)

Set operations **union**, **intersect**, and **except**

- Each of the above operations automatically eliminates duplicates

To retain all duplicates use the

- **union all**,
- **intersect all**
- **except all**.

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

Null Values

It is possible for tuples to have a null value, denoted by **null**, for some of their attributes

null signifies an unknown value or that a value does not exist.

The result of any arithmetic expression involving **null** is **null**

- Example: 5 + **null** returns **null**

The predicate **is null** can be used to check for null values.

- Example: Find all instructors whose salary is null.

```
select name
from instructor
where salary is null
```

The predicate **is not null** succeeds if the value on which it is applied is not null.

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

Null Values (Cont.)

SQL treats as **unknown** the result of any comparison involving a null value (other than predicates **is null** and **is not null**).

- Example: *5* < **null** or **null** <> **null** or **null** = **null**

The predicate in a **where** clause can involve Boolean operations (**and**, **or**, **not**); thus the definitions of the Boolean operations need to be extended to deal with the value **unknown**.

- **and** : (*true* **and** *unknown*) = *unknown*,
  (*false* **and** *unknown*) = *false*,
  (*unknown* **and** *unknown*) = *unknown*
- **or**: (*unknown* **or** *true*) = *true*,
  (*unknown* **or** *false*) = *unknown*
  (*unknown* **or** *unknown*) = *unknown*

Result of **where** clause predicate is treated as *false* if it evaluates to *unknown*

©2022 - Tim Pengajar IF2240 Basis Data
KNOWLEDGE & SOFTWARE ENGINEERING

Latihan-1

1. Tampilkan data ID dan nama student yang berasal dari departemen “Comp. Sci.”.
2. Tampilkan data instruktur dari departemen “Comp. Sci” yang memiliki gaji antara 7.000.000 hingga 12.000.000 terurut mengecil berdasarkan gaji.
3. Tampilkan data student dengan nama berawalan “Budi” dan tediri dari sekurang-kurangnya 2 kata.
4. Tampilkan daftar mata kuliah (kode, nama, sks) yang diambil oleh student dengan ID “13521400” pada semester 2 tahun 2022.
5. Tampilkan daftar seluruh kelas yang ditawarkan oleh departemen “Comp. Sci.” pada semester 2 tahun 2022. Informasi yang ingin ditampilkan meliputi kode kuliah, nama, jumlah sks, nomor kelas (sec_id), serta nama dari instruktur kelas tersebut. Daftar terurut berdasarkan kode kuliah dan nomor kelas.

©2022 - Tim Pengajar IF2240 Basis Data