SQL-The Relational Database Standard-III

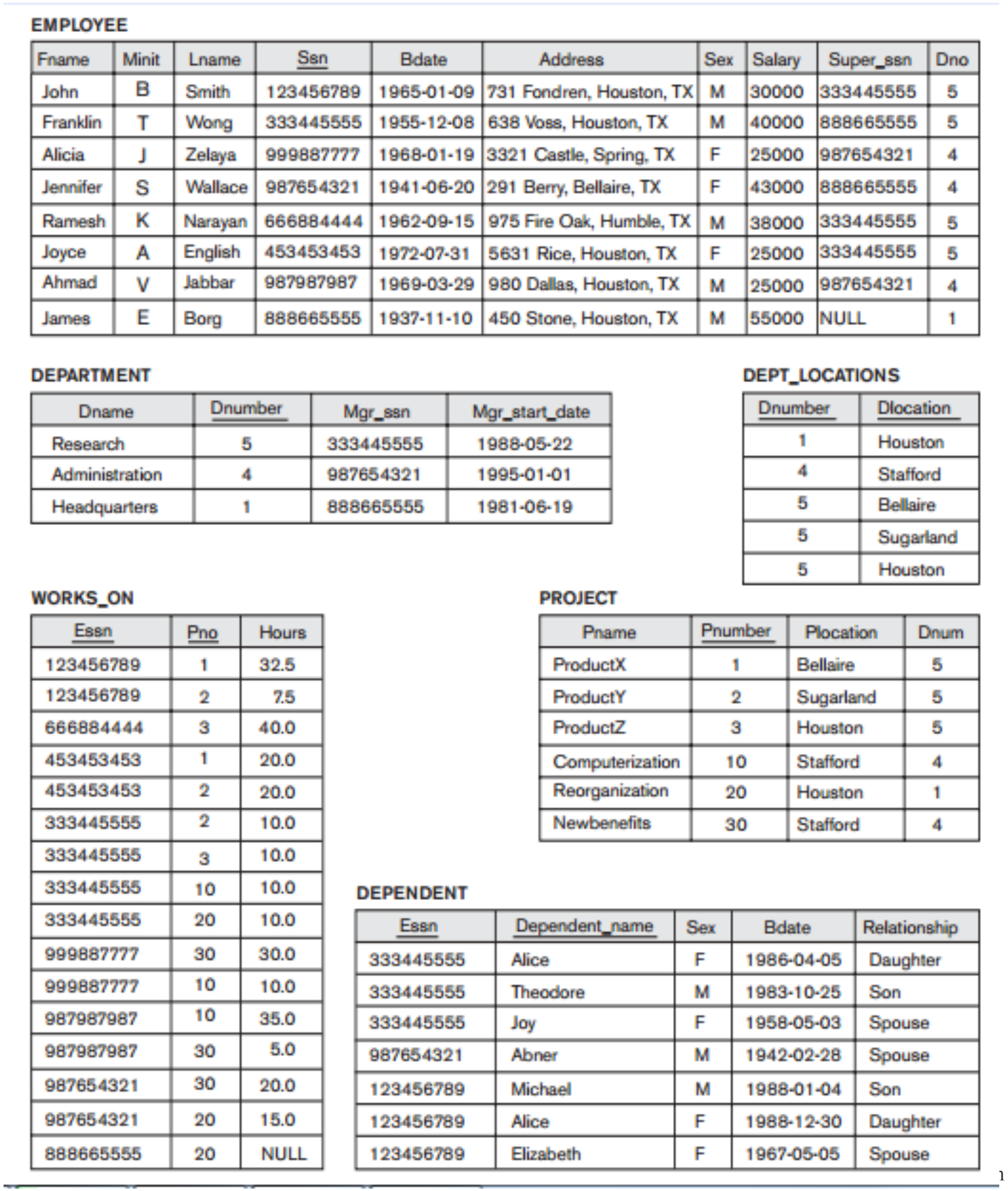
政治大學

資訊科學系

沈錳坤

Summary of SQL Queries

```
SELECT      <attribute list>
FROM<table list>
[WHERE      <condition>]
[GROUP BY <grouping attribute(s)>]
[HAVING     <group condition>]
[ORDER BY <attribute list>]
```



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Update in SQL

Updates in SQL

- three SQL commands to modify the database
  - INSERT
  - DELETE
  - UPDATE

Updates in SQL (cont.)

- **E.g.** Insert a tuple for a new EMPLOYEE for whom we only know the FNAME, LNAME, and SSN attributes.

  INSERT INTO EMPLOYEE (FNAME, LNAME, SSN)

  VALUES ('Richard', 'Marini', '653298653')

- The constraints specified in the DDL commands are automatically enforced by the DBMS when updates are applied to the database

U3A

- Insertion of multiple tuples resulting from a query into a relation

<E.g.> to create a temporary table that has the name, number of employees, and total salaries for each department.

```
U3A:    CREATE TABLE DEPTS_INFO
        (DEPT_NAME   VARCHAR(10),
        NO_OF_EMPS  INTEGER,
        TOTAL_SAL    INTEGER);

INSERT INTO
    DEPTS_INFO (DEPT_NAME, NO_OF_EMPS, TOTAL_SAL)
    SELECT  DNAME, COUNT (*), SUM (SALARY)
    FROM        DEPARTMENT, EMPLOYEE
    WHERE       DNUMBER = DNO
    GROUP BY  DNAME ;
```

DELETE

- Removes tuples from a relation
- Includes a WHERE-clause to select the tuples to be deleted
- The number of tuples deleted depends on the number of tuples in the relation that satisfy the WHERE-clause
- Tuples are deleted from only one table at a time (unless CASCADE is specified on a referential integrity constraint)
- A missing WHERE-clause specifies that all tuples in the relation are to be deleted; the table then becomes an empty table
- Referential integrity should be enforced

U4A/U4B/U4C/U4D

```
U4A:   DELETE FROM EMPLOYEE
       WHERE        LNAME = 'Brown'

U4B:   DELETE FROM EMPLOYEE
       WHERE        SSN = '123456789'

U4C:   DELETE FROM  EMPLOYEE
       WHERE  DNO  IN
                ( SELECT DNUMBER
                  FROM   DEPARTMENT
                  WHERE DNAME = 'Research')

U4D:   DELETE FROM EMPLOYEE
```

delete all records

UPDATE

- Used to modify attribute values of one or more selected tuples
- A WHERE-clause selects the tuples to be modified
- An additional SET-clause specifies the attributes to be modified and their new values
- Each command modifies tuples in the same relation
- Referential integrity should be enforced

U5/U6

U5: Change the location and controlling department number of project number 10 to 'Bellaire' and 5, respectively

```
UPDATE   PROJECT
SET      PLOCATION = 'Bellaire', DNUM = 5
WHERE    PNUMBER = 10
```

U6: Give all employees in the 'Research' department a 10% raise in salary.

```
UPDATE   EMPLOYEE
SET      SALARY = SALARY * 1.1
WHERE  DNO  IN
        (  SELECT    DNUMBER
           FROM      DEPARTMENT
           WHERE     DNAME = 'Research' )
```

View in SQL

Relational Views in SQL

- A view is a single virtual table that is derived from other tables –

  (1) base tables or

  (2) previously defined views

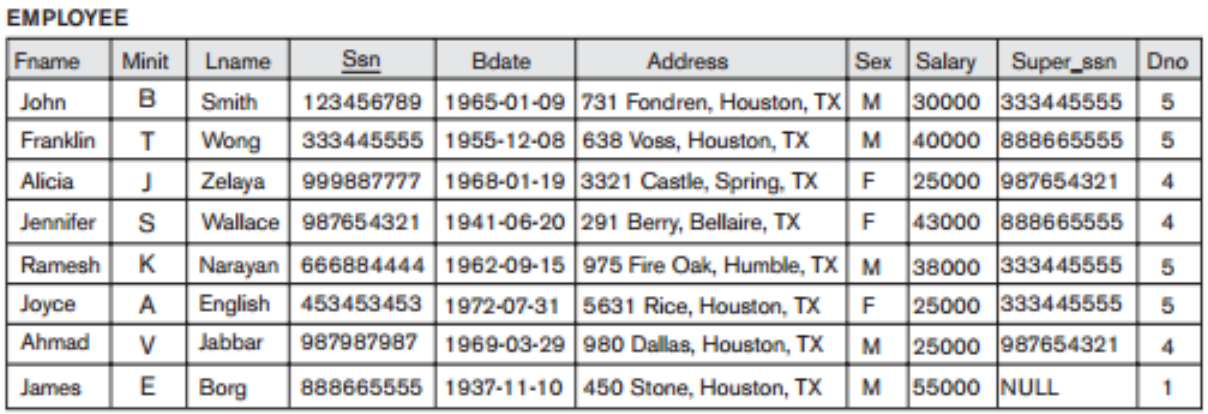
- A view does not necessarily exist in physical form, which limits the possible update operations that can be applied to views
- There are no limitations on querying a view
- **CREATE VIEW** command specify a view by specifying a (virtual) table name and a defining query
- The view attribute names can be inherited from the attribute names of the tables in the defining query

Relational Views in SQL (cont.)

- One advantages of a view is to simplify the specification of queries.
- Views can also be used as a security and authorization mechanism
- DBMS responsible for keeping the view up-to-date if the base tables on which the view is defined are modified
- It is the responsibility of DBMS, not the user, to make sure that the view is up to date
- View is not realized at the time of view definition, but at the time we specify a query on the view

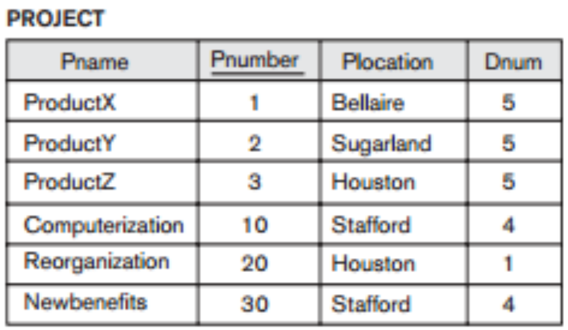
V1

- **CREATE** VIEW WORKS_ON1 **AS**
  SELECT FNAME, LNAME, PNAME, HOURS
  FROM EMPLOYEE, PROJECT, WORKS_ON
  WHERE SSN**=**ESSN **AND** PNO**=**PNUMBER **;**



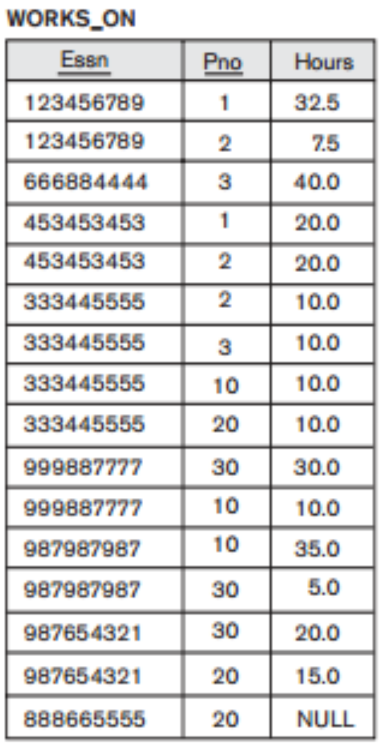
EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1



PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4



WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

QUERIES ON VIEWS

- Retrieve the last name and first name of all employees who work on 'ProjectX'.

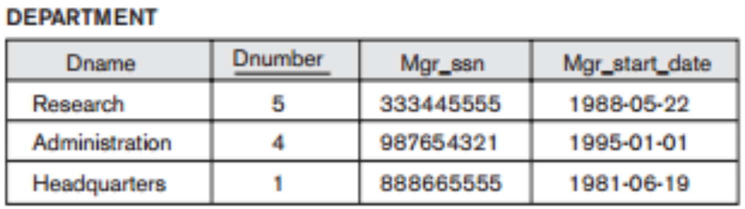
```
QV1: SELECT   PNAME, FNAME, LNAME
     FROM     WORKS_ON1
     WHERE    PNAME='ProjectX' ;
```

```
* CREATE VIEW      WORKS_ON1  AS
  SELECT           FNAME, LNAME, PNAME, HOURS
  FROM             EMPLOYEE, PROJECT, WORKS_ON
  WHERE            SSN=ESSN AND PNO=PNUMBER ;
```

V2

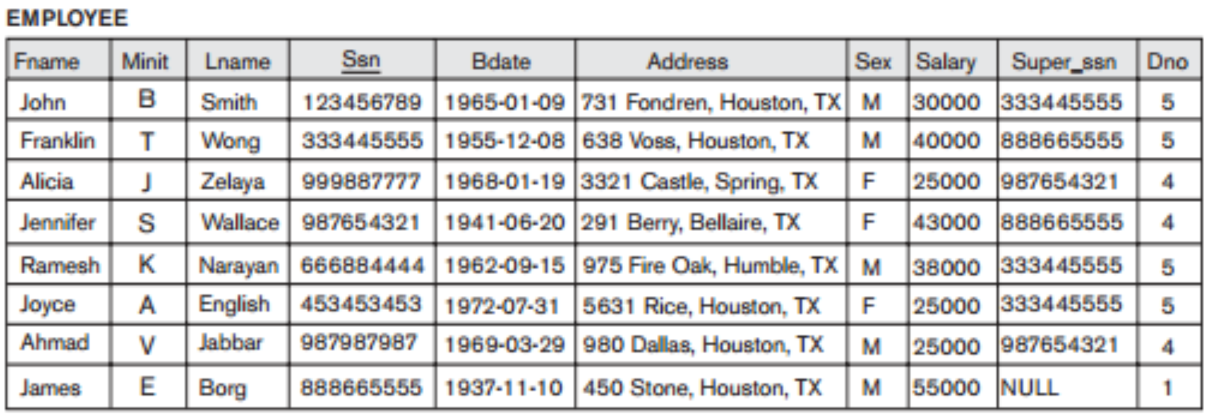
```
CREATE VIEW DEPT_INFO (DEPT_NAME, NO_OF_EMPS, TOTAL_SAL)
AS
        SELECT  DNAME, COUNT(*), SUM(SALARY)
        FROM    DEPARTMENT, EMPLOYEE
        WHERE   DNUMBER=DNO
        GROUP BY DNAME ;
```

specify column names



DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Drop View

A view is removed using the **DROP VIEW** command

V1A: **DROP VIEW** WORKS_ON1 **;**

V2A: **DROP VIEW** DEPT_INFO **;**

View 是 Virtual Table，

DBMS 如何實作 View 的功能？

View Implementation

- Two main approaches
  - Query modification
    - Modifying view query into a query on the underlying base tables
    - Disadvantage: inefficient for views of complex queries
  - View materialization
    - Physically creating a temporary view table
    - Incremental update view when base tables are updated
    - If view is not queried for a period of time, system automatically remove the physical view table

View Implementation (cont.)

- Query modification: modifying view query into a query on the underlying base tables

create view

```
CREATE VIEW      WORKS_ON1  AS
SELECT  FNAME, LNAME, PNAME, HOURS
FROM     EMPLOYEE, PROJECT, WORKS_ON
WHERE   SSN = ESSN AND PNO = PNUMBER ;
```

query view

```
SELECT  PNAME, FNAME, LNAME
FROM     WORKS_ON1
WHERE   PNAME = 'ProjectX' ;
```

```
SELECT  PNAME, FNAME, LNAME
FROM     EMPLOYEE, PROJECT, WORKS_ON
WHERE   SSN = ESSN AND PNO = PNUMBER
        AND PNAME='ProjectX' ;
```

Updating of Views

- A view update operation may be mapped in multiple ways to update operations on the defining base relations - ambiguity
- updating views is still an active research area

Ex: To update the WORKS_ON1 view by modifying the PNAME attribute of 'John Smith' from 'ProductX' to 'ProductY'.

```
UV1:  UPDATE  WORKS_ON1
      SET     PNAME = 'ProductY'
      WHERE   LNAME='Smith' AND
              FNAME='John' AND
              PNAME='ProductX'
```

View Update 如何對應到其 Base Table 的 Update？

John Smith 參與的計畫不變，只是改名為 Product Y

OR

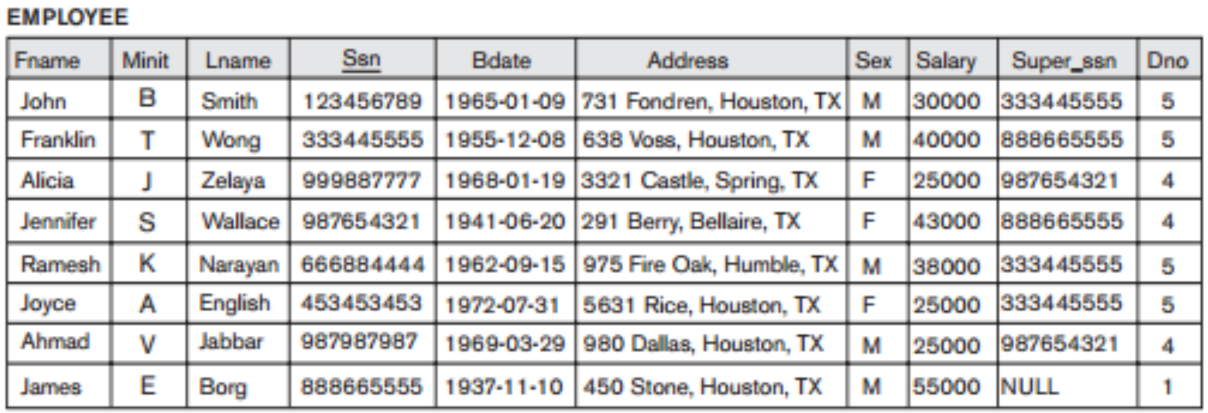
John Smith 參與的計畫改變，改參與Product Y 計畫？

Updating of Views (cont.)

- Updates on the base relations to give the desired update on the view
- Two possibilities

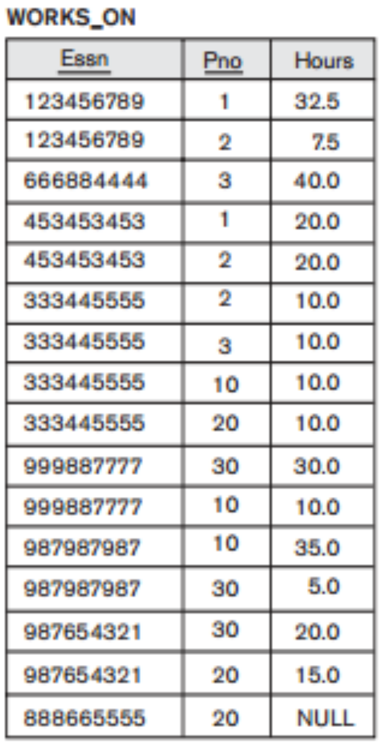
(1) Change the name of the 'ProductX' tuple in the PROJECT relation to 'ProductY‘

```
UPDATE    PROJECT
SET       PNAME = 'ProductY'
WHERE     PNAME = 'ProductX'
```



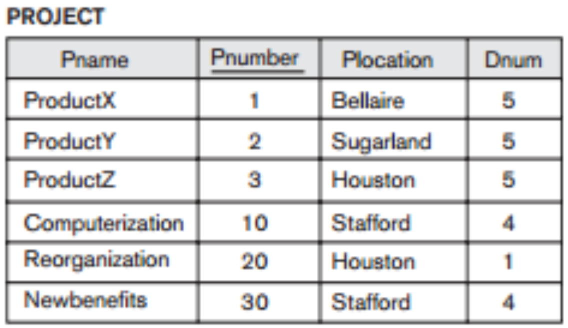
EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1



WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL



PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Updating of Views (cont.)

(2) Relate 'John Smith' to the 'ProductY' PROJECT tuple in place of the 'ProductX' PROJECT tuple

```
UPDATE   WORKS_ON
SET PNO = ( SELECT   PNUMBER
            FROM     PROJECT
           WHERE PNAME = 'ProductY’ )
 WHERE  ESSN = ( SELECT  SSN
                  FROM     EMPLOYEE
                  WHERE LNAME = 'Smith'
                        AND FNAME = 'John’ )
    AND  PNO =  ( SELECT PNUMBER
                   FROM PROJECT
                   WHERE PNAME = 'ProductX’ )
```

the project # of ProductY

John Smith

works on ProductX

Updating of Views (cont.)

- Some view updates may not make much sense

```
UV2: UPDATE   DEPT_INFO
         SET TOTAL_SAL = 100000   如何分配？
         WHERE DNAME = 'Research' ;
```

- In general, we cannot guarantee that any view can be updated
- A view update is unambiguous only if one update on the base relations can accomplish the desired update effect on the view
- If a view update can be mapped to more than one update on the underlying base relations, we must have a certain procedure to choose the desired update

Updating of Views (cont.)

- A view with a single defining table is updatable if the view attributes contain the primary key
- Views defined on multiple tables using joins are generally not updatable
- Views defined using grouping and aggregate functions are not updatable

WITH

- 假設薪水最高的員工不只一位，列出所有這些員工的SSN。
  –
```
CREATE VIEW Max_Salary(Value)
 AS SELECT Max(Salary)
 FROM Employee;

 SELECT SSN
 FROM Employee, Max_Salary
 WHERE Employee.Salary = Max_Salary.Value;
```
  –
```
WITH Max_Salary(Value)
          AS SELECT Max(Salary)
             FROM Employee
 SELECT SSN
 FROM Employee, Max_Salary
 WHERE Employee.Salary = Max_Salary.Value;
```

Allows you to give the sub-query block a name

Assertion

semantic integrity constraint

Constraint: salary of an employee must not be greater than the salary of the manager of the department that the employee works for (MySQL does not support assertion & domain)

```
CREATE ASSERTION SALARY_CONSTRAINT
CHECK (NOT EXISTS (SELECT * FROM EMPLOYEE E,
                          EMPLOYEE M, DEPARTMENT D
                    WHERE E.SALARY > M.SALARY AND
              E.DNO=D.DNUMBER AND D.MGRSSN=M.SSN));
```

```
CREATE DOMAIN D_NUM AS INTEGER
CHECK (D_NUM > 0 AND D_NUM < 21);
```

essentially a data type w/ constraints than can be used in multiple tables

Additional Features of SQL

Additional Features of SQL

- Granting, revoking of privileges to users to access certain relations e.g. GRANT SELECT, INSERT ON 'ALICE'@'localhost'
  REVOKE INSERT ON employees FROM 'Alice'@'localhost'
- Embed SQL in programming language C, C++, COBOL, PASCAL based on cursor
- Transaction control granularity for concurrency control, recovery → an indivisible unit of work
- Physical database design parameters, file structures for relations, access paths as indexes

A tomic : All changes to the data must be performed or not at all

C onsistent : data must be consistent before & after the transaction

I solated : no other process can change the data during the transaction

D urable : the changes of the transaction must persist

Creating Indexes in SQL

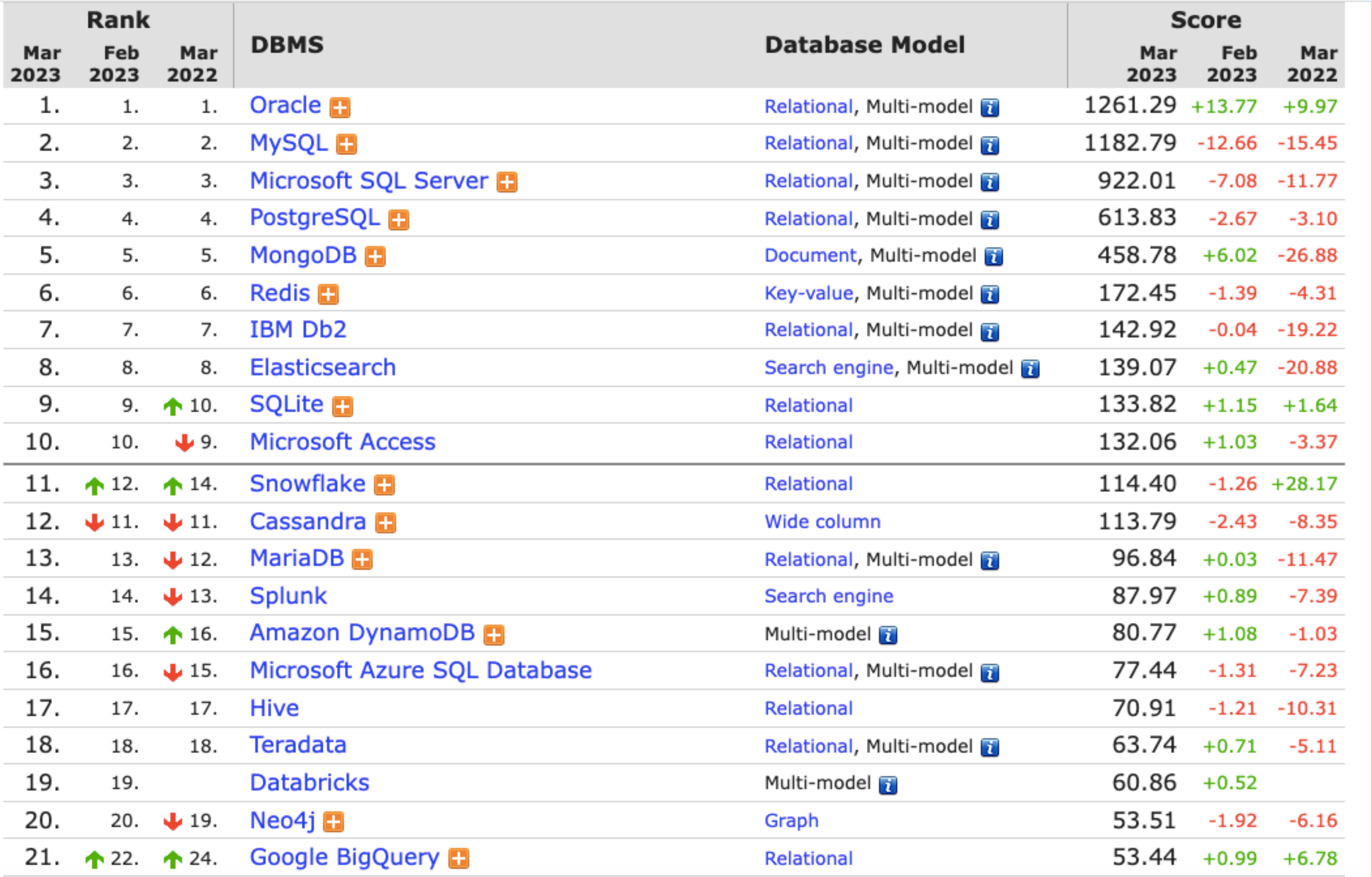
- An SQL base relation generally corresponds to a stored file
- Statements can create and drop indexes on base relations
- These statements have been removed from SQL2 because they specify physical access paths - not logical concepts
- One or more indexing attributes are specified for each index
- **CREATE INDEX** statement is used
- Each index is given an *index name*

index name

I1: **CREATE INDEX** LNAME_INDEX

ON EMPLOYEE(LNAME);

relation attribute list



Rank			DBMS	Database Model	Score		
Mar 2023	Feb 2023	Mar 2022			Mar 2023	Feb 2023	Mar 2022
1.	1.	1.	Oracle	Relational, Multi-model	1261.29	+13.77	+9.97
2.	2.	2.	MySQL	Relational, Multi-model	1182.79	-12.66	-15.45
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	922.01	-7.08	-11.77
4.	4.	4.	PostgreSQL	Relational, Multi-model	613.83	-2.67	-3.10
5.	5.	5.	MongoDB	Document, Multi-model	458.78	+6.02	-26.88
6.	6.	6.	Redis	Key-value, Multi-model	172.45	-1.39	-4.31
7.	7.	7.	IBM Db2	Relational, Multi-model	142.92	-0.04	-19.22
8.	8.	8.	Elasticsearch	Search engine, Multi-model	139.07	+0.47	-20.88
9.	9.	↑ 10.	SQLite	Relational	133.82	+1.15	+1.64
10.	10.	↓ 9.	Microsoft Access	Relational	132.06	+1.03	-3.37
11.	↑ 12.	↑ 14.	Snowflake	Relational	114.40	-1.26	+28.17
12.	↓ 11.	↓ 11.	Cassandra	Wide column	113.79	-2.43	-8.35
13.	13.	↓ 12.	MariaDB	Relational, Multi-model	96.84	+0.03	-11.47
14.	14.	↓ 13.	Splunk	Search engine	87.97	+0.89	-7.39
15.	15.	↑ 16.	Amazon DynamoDB	Multi-model	80.77	+1.08	-1.03
16.	16.	↓ 15.	Microsoft Azure SQL Database	Relational, Multi-model	77.44	-1.31	-7.23
17.	17.	17.	Hive	Relational	70.91	-1.21	-10.31
18.	18.	18.	Teradata	Relational, Multi-model	63.74	+0.71	-5.11
19.	19.		Databricks	Multi-model	60.86	+0.52	
20.	20.	↓ 19.	Neo4j	Graph	53.51	-1.92	-6.16
21.	↑ 22.	↑ 24.	Google BigQuery	Relational	53.44	+0.99	+6.78

(cf: https://db-engines.com/en/ranking)

Database Programming

- ♦ Approaches to DB programming
  - Embedding DB commands in a general-purpose programming language
    - Embedded SQL
    - Dynamic SQL : SQL statements are constructed at runtime
    - SQLJ : a standard that allows embedding SQL into JAVA
  - Using a library of DB functions (API)
    - SQL/CLI (Call Level Interface, ODBC) → Open Database Connectivity: A widely used implementation of SQL/CLI
    - JDBC : Java Database Connectivity
  - Designing a brand-new language
    - Oracle PL/SQL : Oracle's procedural extension for SQL and Oracle's relational database.

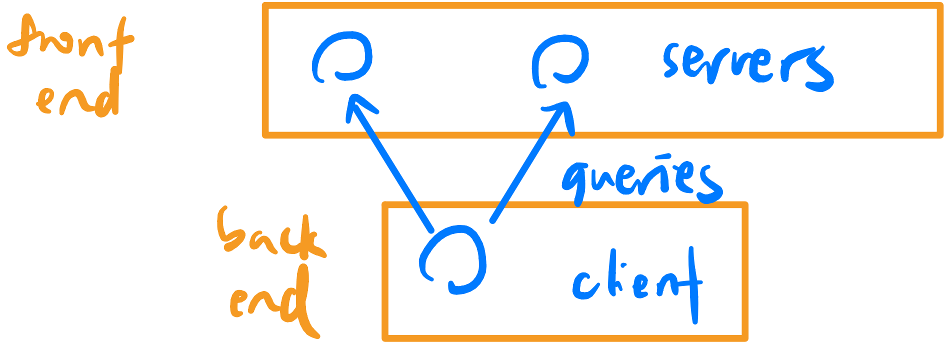
Impedance mismatch is a term used in computer science to describe the problem that arises when two systems or components that are supposed to work together have different data models, structures, or interfaces that make communication difficult or inefficient.

Impedance Mismatch

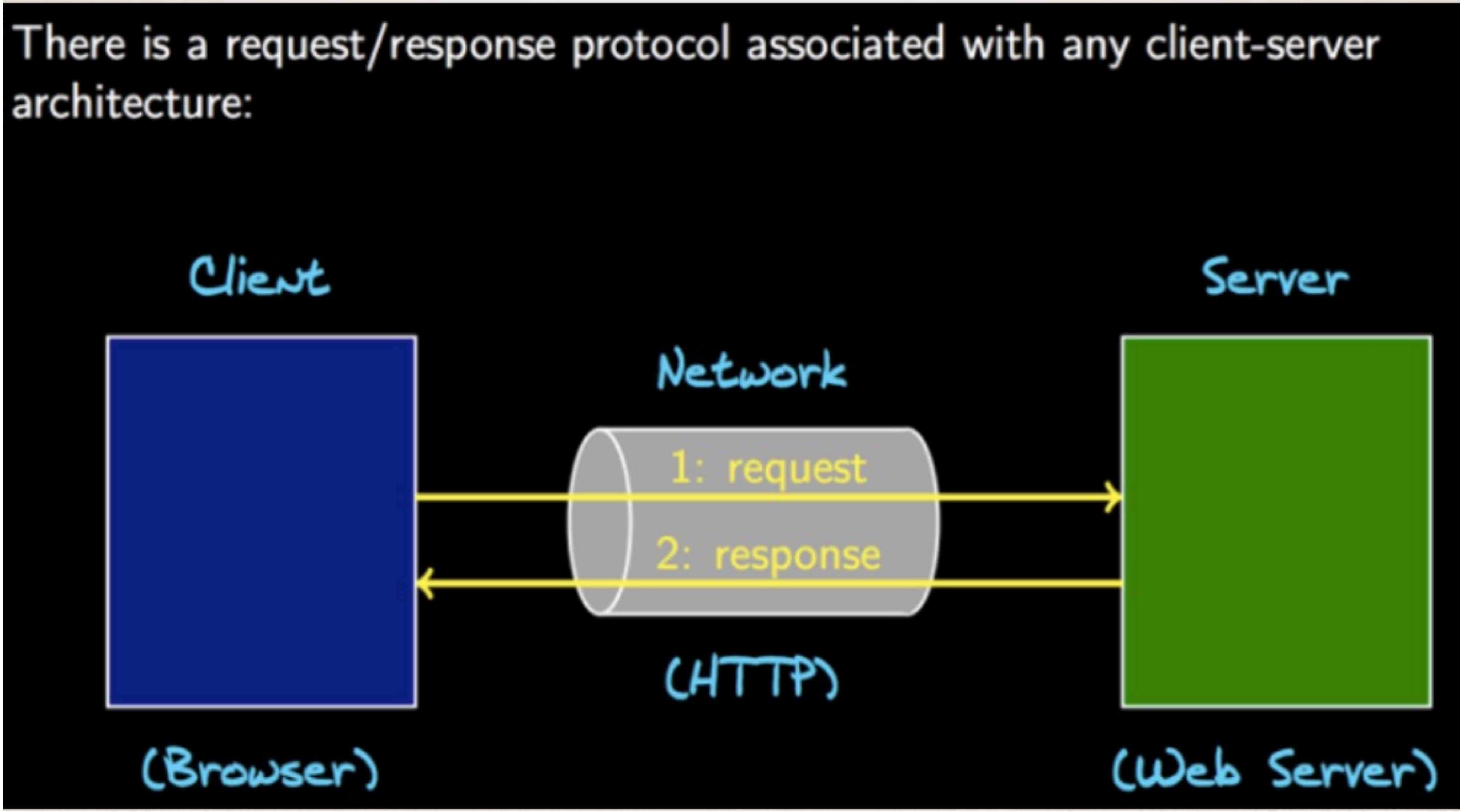
- Problem occurs because of difference between the DB model & PL model → Programming Language
  - Attribute data type vs. data type of PL: binding
  - Mapping between query result data structure & data structure in PL: cursor (iterator variable) is used to loop over the tuples in a query result
  - Impedance mismatch is less a problem when a special DB PL is designed.

Typical Sequence of Interaction in DB Programming

- Client/server model
  - Client program handles the logic of a software application
  - Client includes some calls to one or more database server to access or update data
  - Front-end vs. Back-end
- Common sequence
  - Client program establish or open a connection to the database server
  - Once the connection is established, program interact with the DB by submitting queries, updates or other commands
  - Close the connection

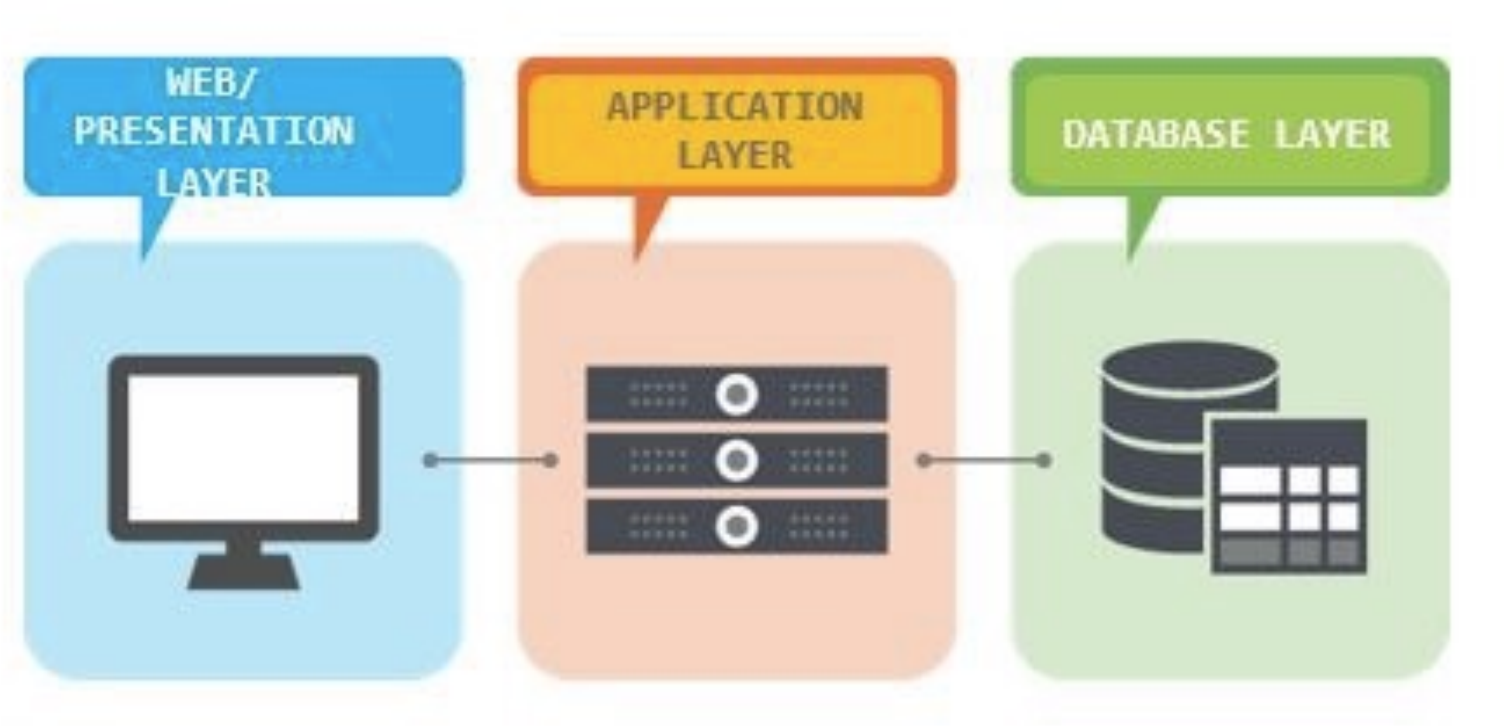


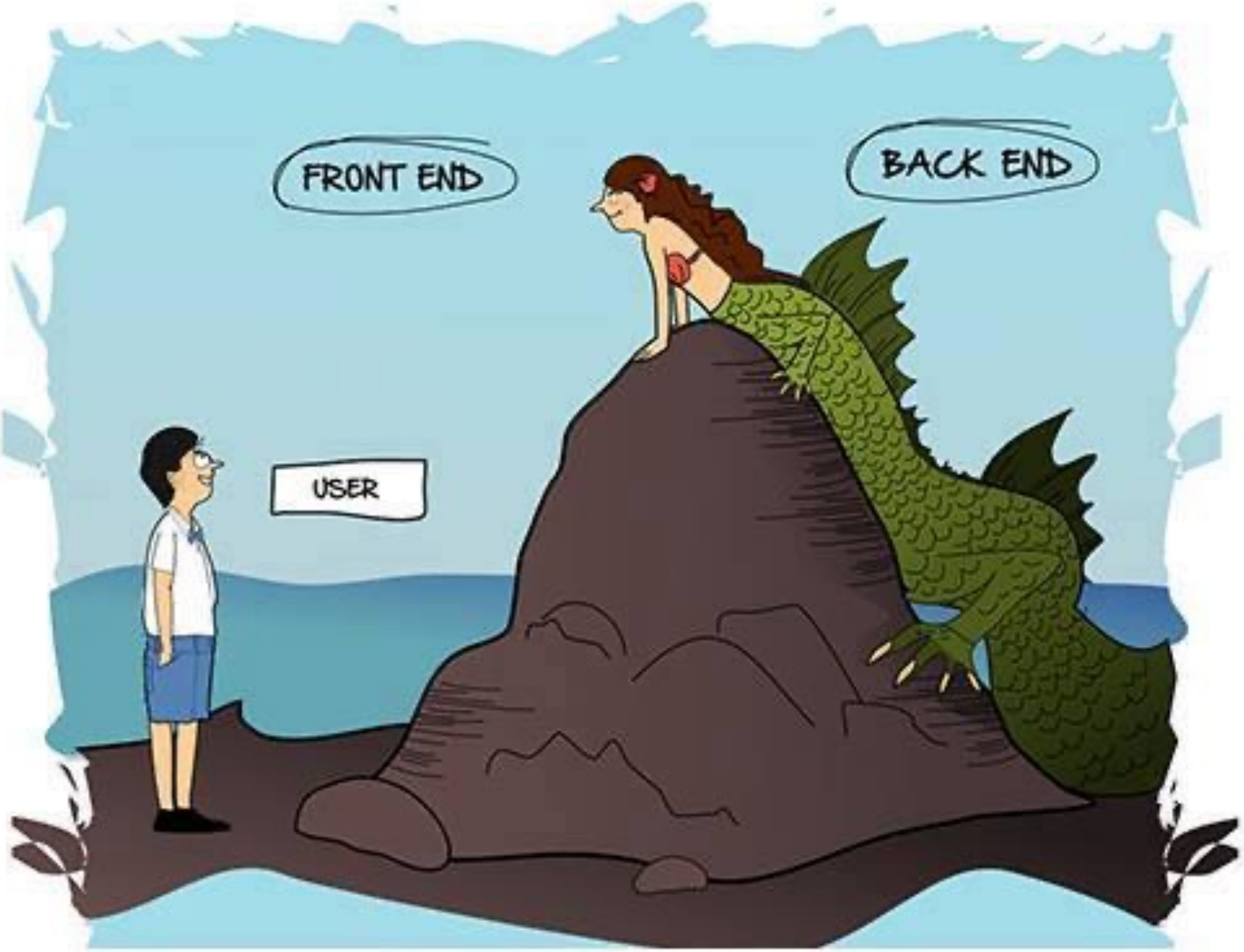
Client-Server



There is a request/response protocol associated with any client-server architecture:

Client-Server (cont.)





Front end vs. Back end.

恰?!我死了?!
!
Well Done
馬面
這輩子太辛苦了
希望可以轉生
異世界開後宮
過勞死的工程師呀…
不好好愛惜自己身體
你在地獄的懲罰就是把
生死簿做成後台管理系統
生死簿
…
瀏覽器都幫
你準備好了
畢竟只有他下來了
Medium Rare
二次死亡
呀呀呀呀呀呀呀呀呀呀呀呀
工程師的壓力有點大

Summary

- SQL
- DDL, DML, View, Indexing, …
- Embedd SQL + Host language
- Syntax

SELECT <attribute list>

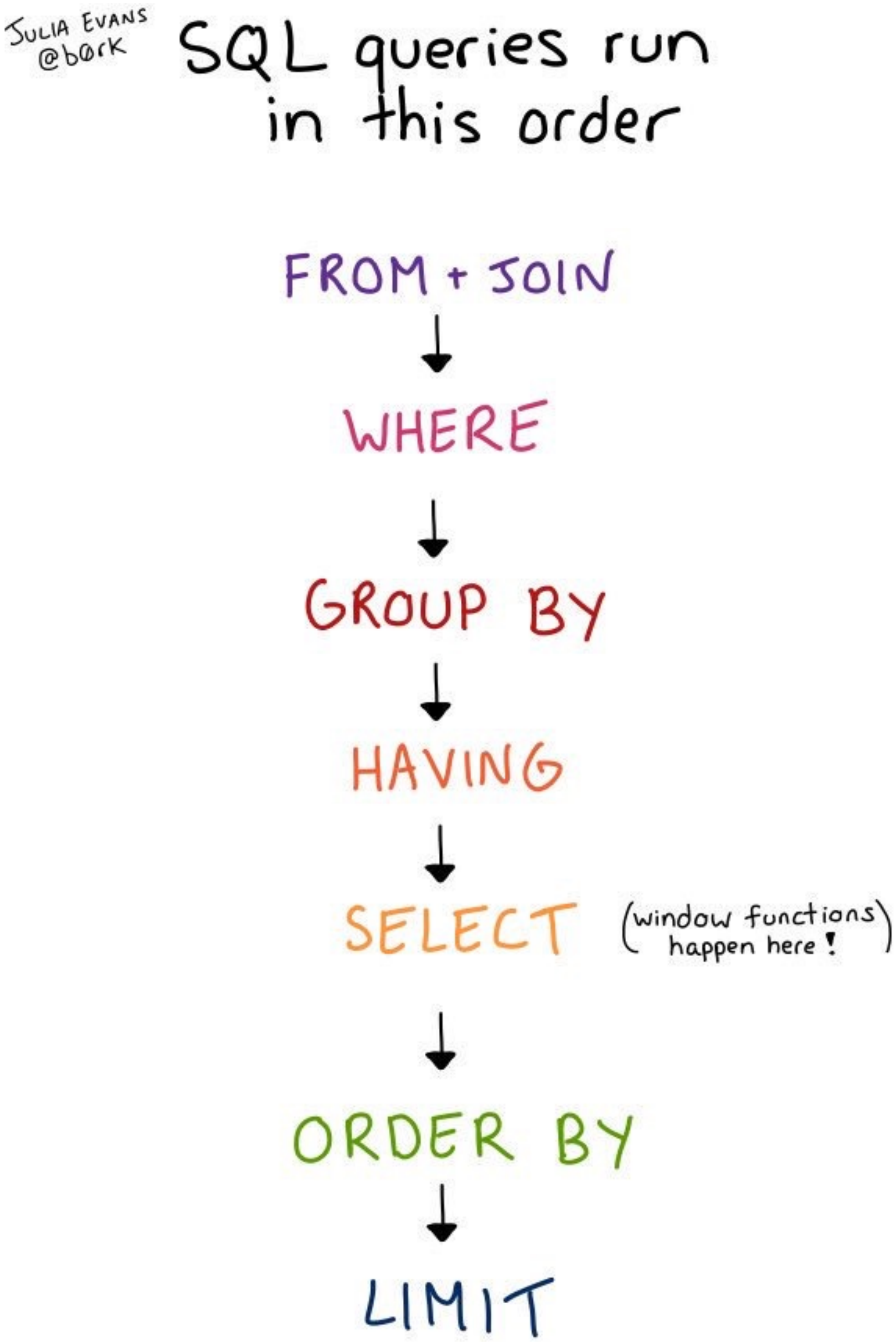
FROM <table list>

[**WHERE** <condition>]

[**GROUP BY** <grouping attribute(s)>]

[**HAVING** <group condition>]

[**ORDER BY** <attribute list>]


JULIA EVANS
@børk
SQL queries run in this order
FROM + JOIN
WHERE
GROUP BY
HAVING
SELECT
(window functions happen here!)
ORDER BY
LIMIT