A crash course on relational database design

Professor Hossein Saiedian

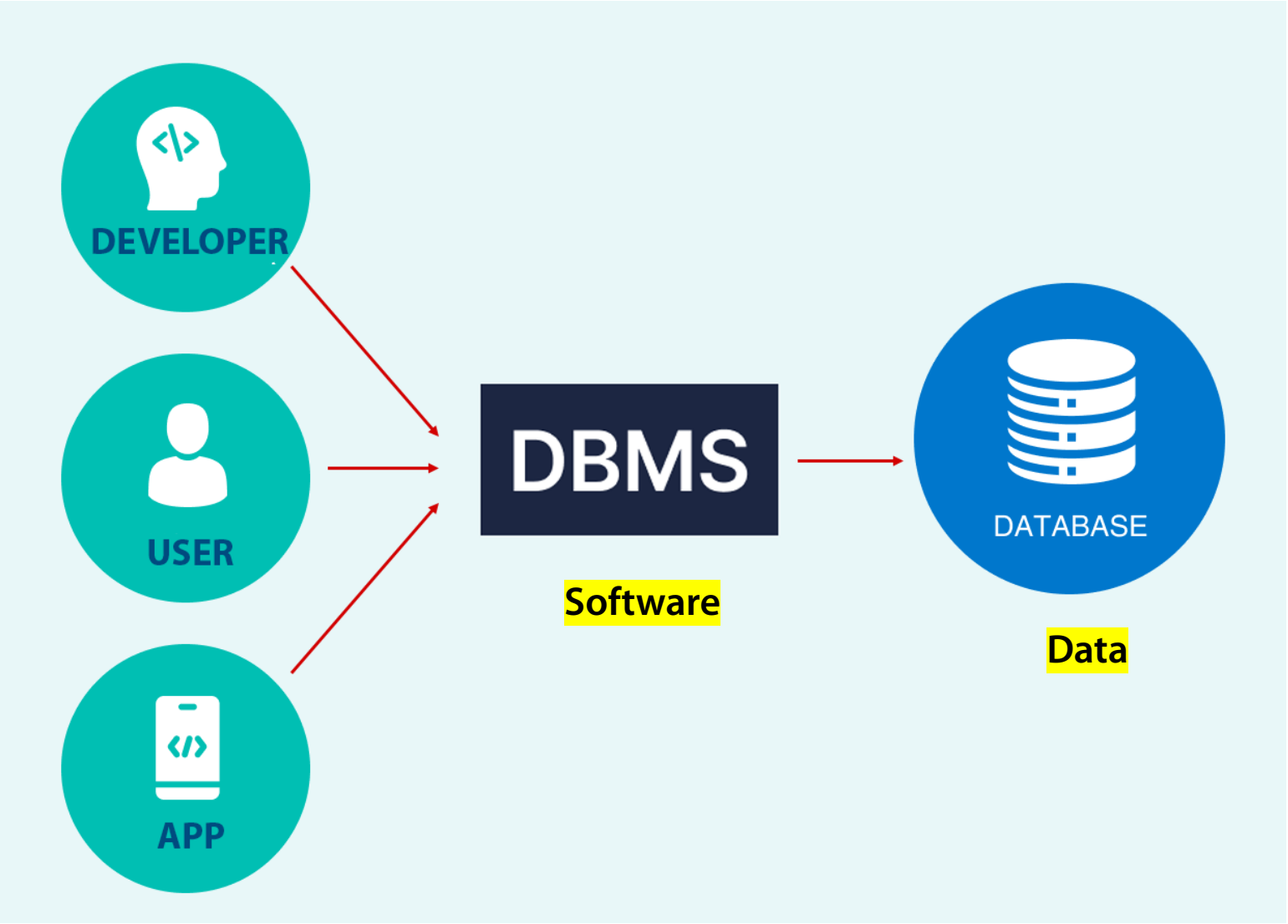
EECS 348: Software Engineering

Fall 2023

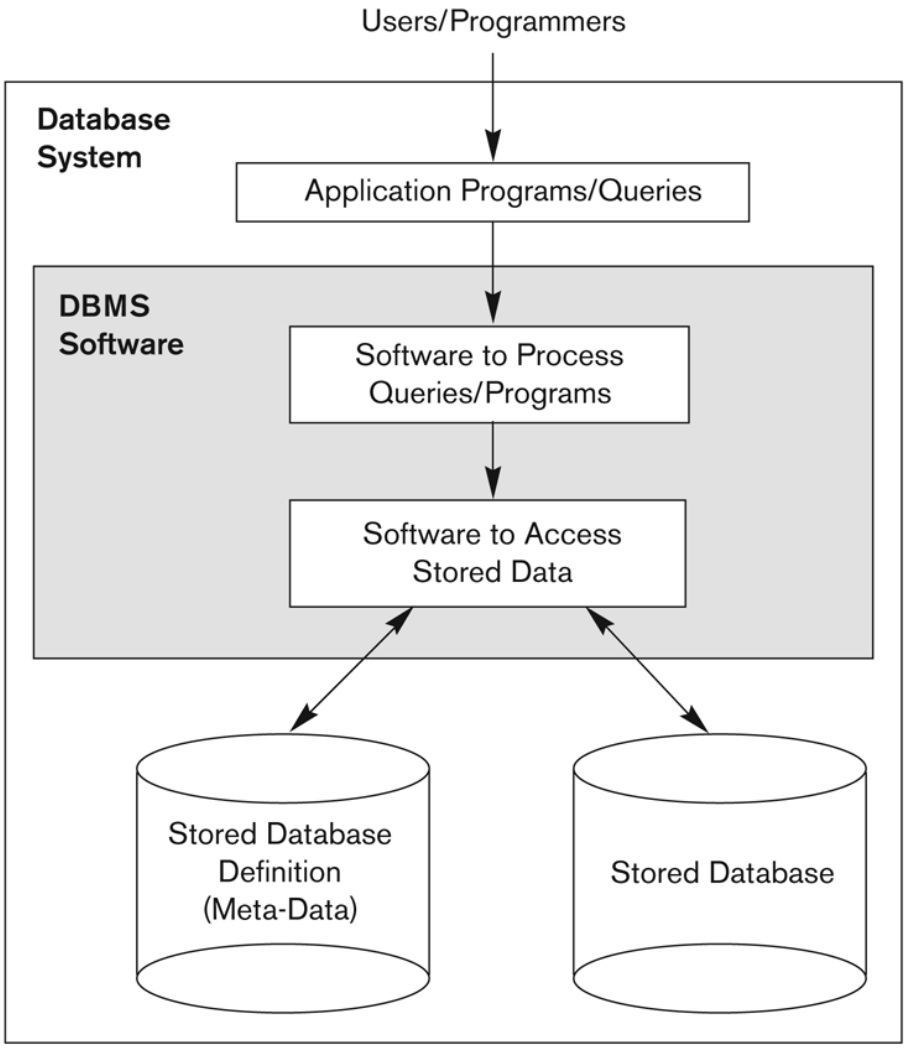
DBMS and a database

- A DBMS is a set of programs
- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access the data
  - Provides an environment that is both convenient and efficient to use
- Database systems are used to manage collections of data that are
  - Relatively large
  - Accessed by multiple users and applications, often at the same time

A simplified architecture for a DBMS



A simplified architecture for a Database



Example of a database

- A UNIVERSITY environment or mini-world

Some mini-world entities	Some mini-world relationships
STUDENTs	SECTIONs are of specific COURSEs
COURSEs	STUDENTs take SECTIONs
SECTIONs (or COURSEs)	COURSEs have prerequisite COURSEs
DEPARTMENTs	INSTRUCTORs teach SECTIONs
INSTRUCTORs	COURSEs are offered by DEPARTMENTs
	STUDENTs major in DEPARTMENTs

- Note: The above entities and relationships are typically expressed in a conceptual data model, such as the UML class diagram or the entity-relationship (ER) model

Example of a database

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

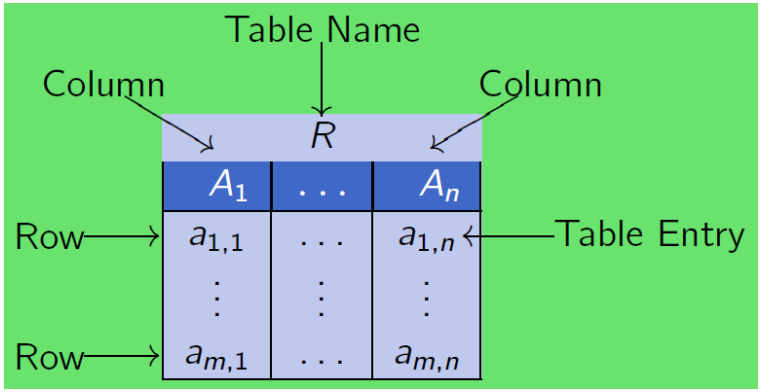
Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310



The relational model

E.F. "Ted" Codd

Columns

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

Rows

Database languages

- Data Definition Language (DDL)
  - Used by the DBA and database designers to specify the conceptual schema of a database; to define views
- Data Manipulation Language (DML)
  - Used to specify database retrievals and updates
  - DML commands can be *embedded* in a general-purpose programming language (e.g., C++)

DBMS interfaces

- Stand-alone query language interfaces (e.g., SQL)
- Programmer interfaces for embedding DML in a PL
- User-friendly interfaces (menu-based, forms-based)
- Natural language: requests in written English
- Combinations of the above

Overview of the DB design process

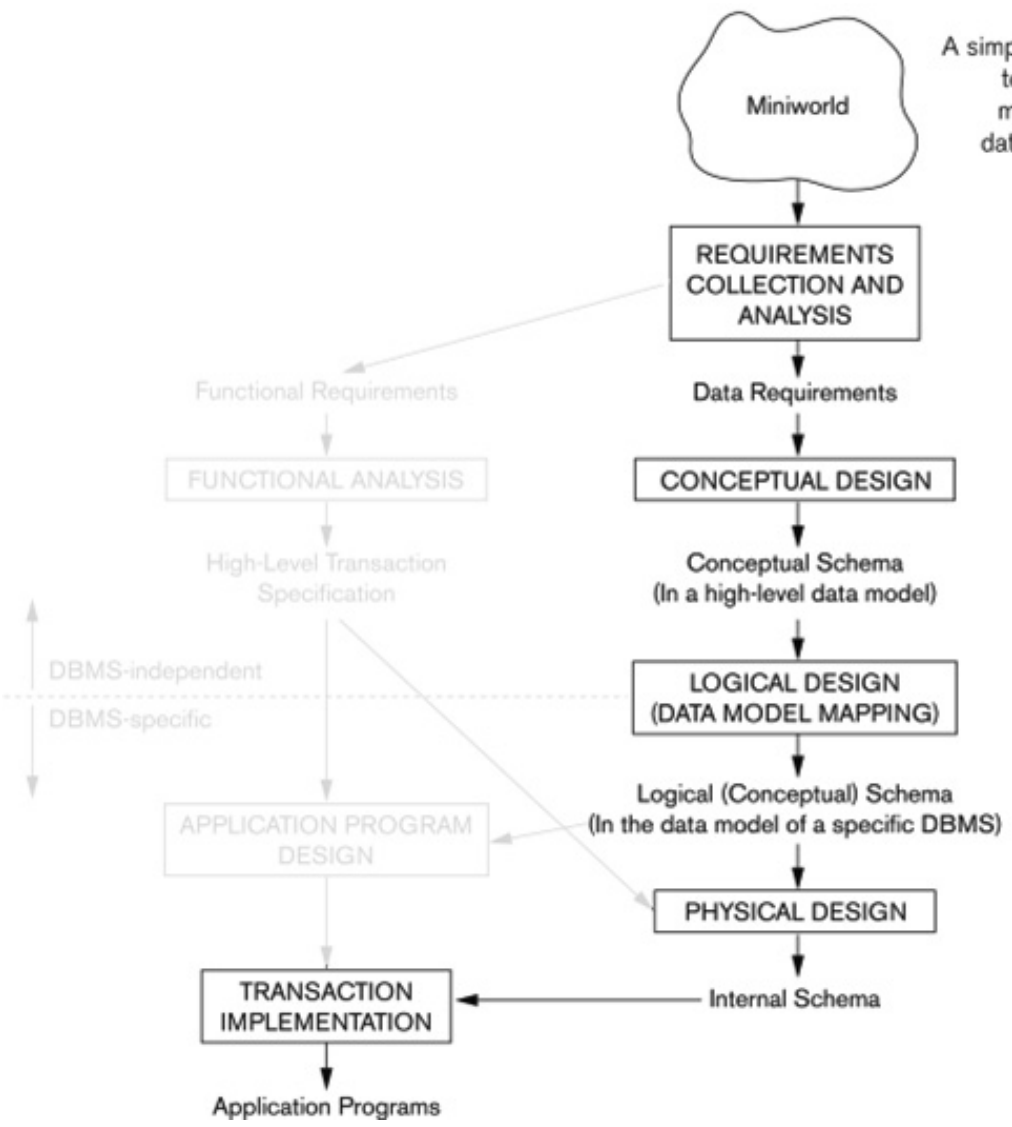


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

A sample database application

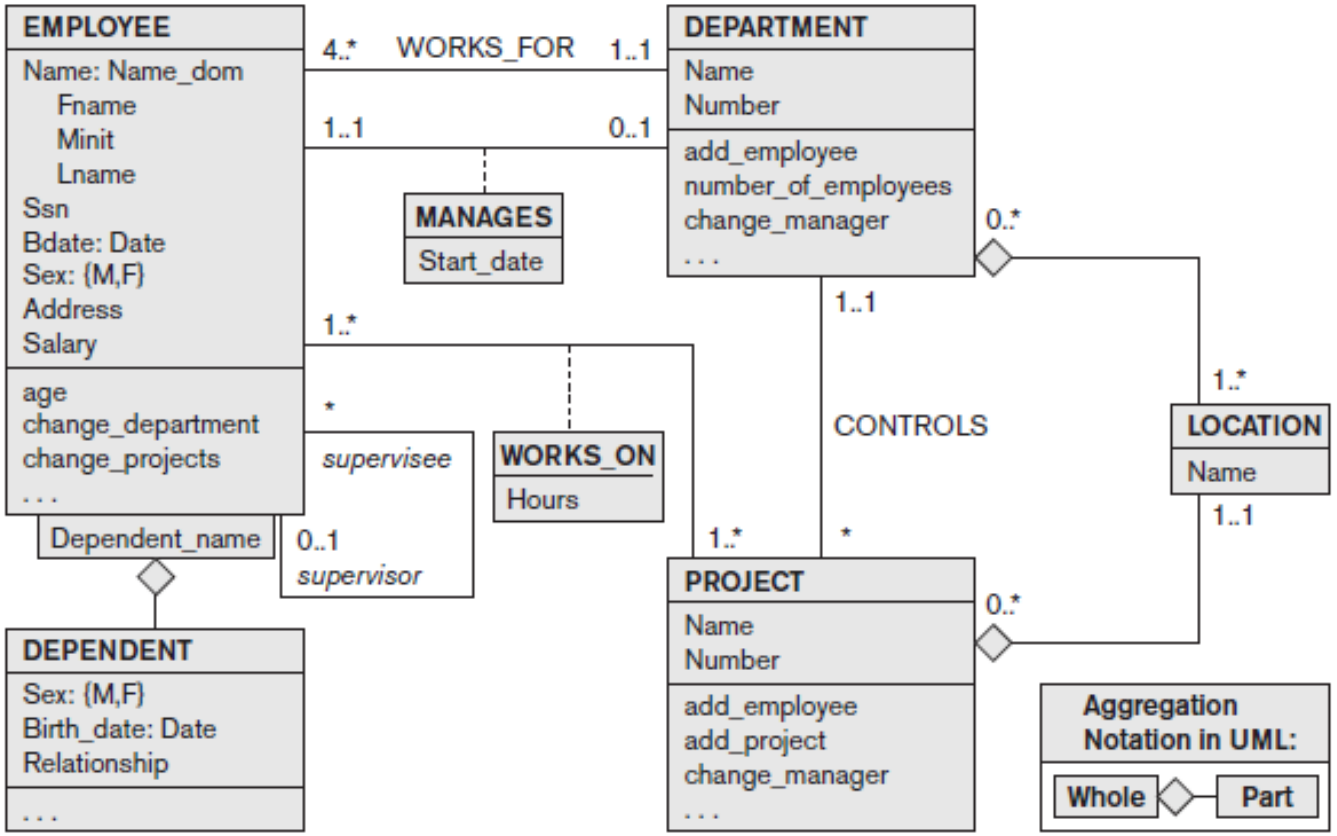
The company is organized into ***departments***. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.

A department controls a number of ***projects***, each of which has a unique name, a unique number, and a single location.

The database will store each ***employee's*** name, Social Security number, address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. It is required to keep track of the current number of hours per week that an employee works on each project, as well as the direct supervisor of each employee (who is another employee).

The database will keep track of the ***dependents*** of each employee for insurance purposes, including each dependent's first name, sex, birth date, and relationship to the employee.

Conceptual modeling (UML)



Overview of the DB design process

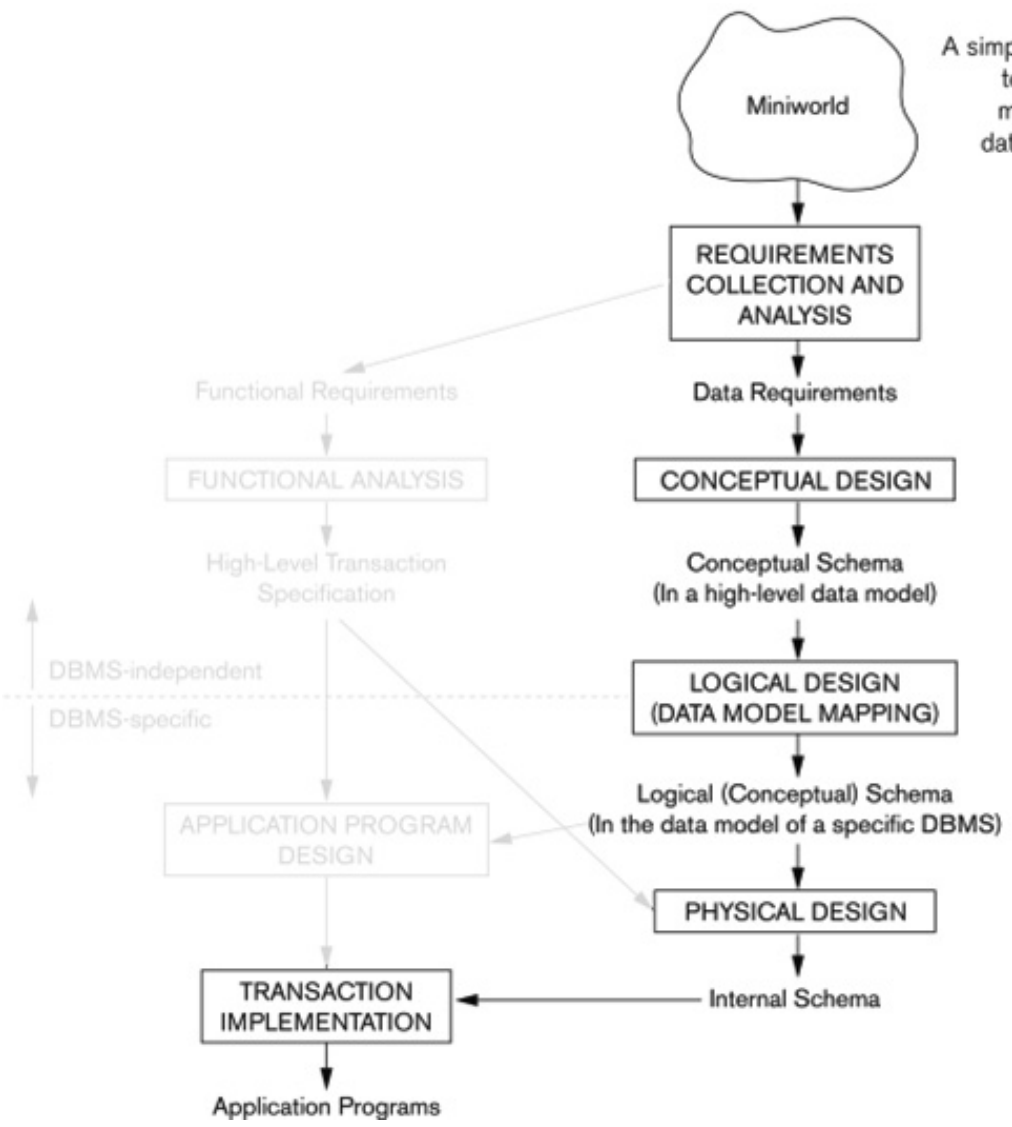
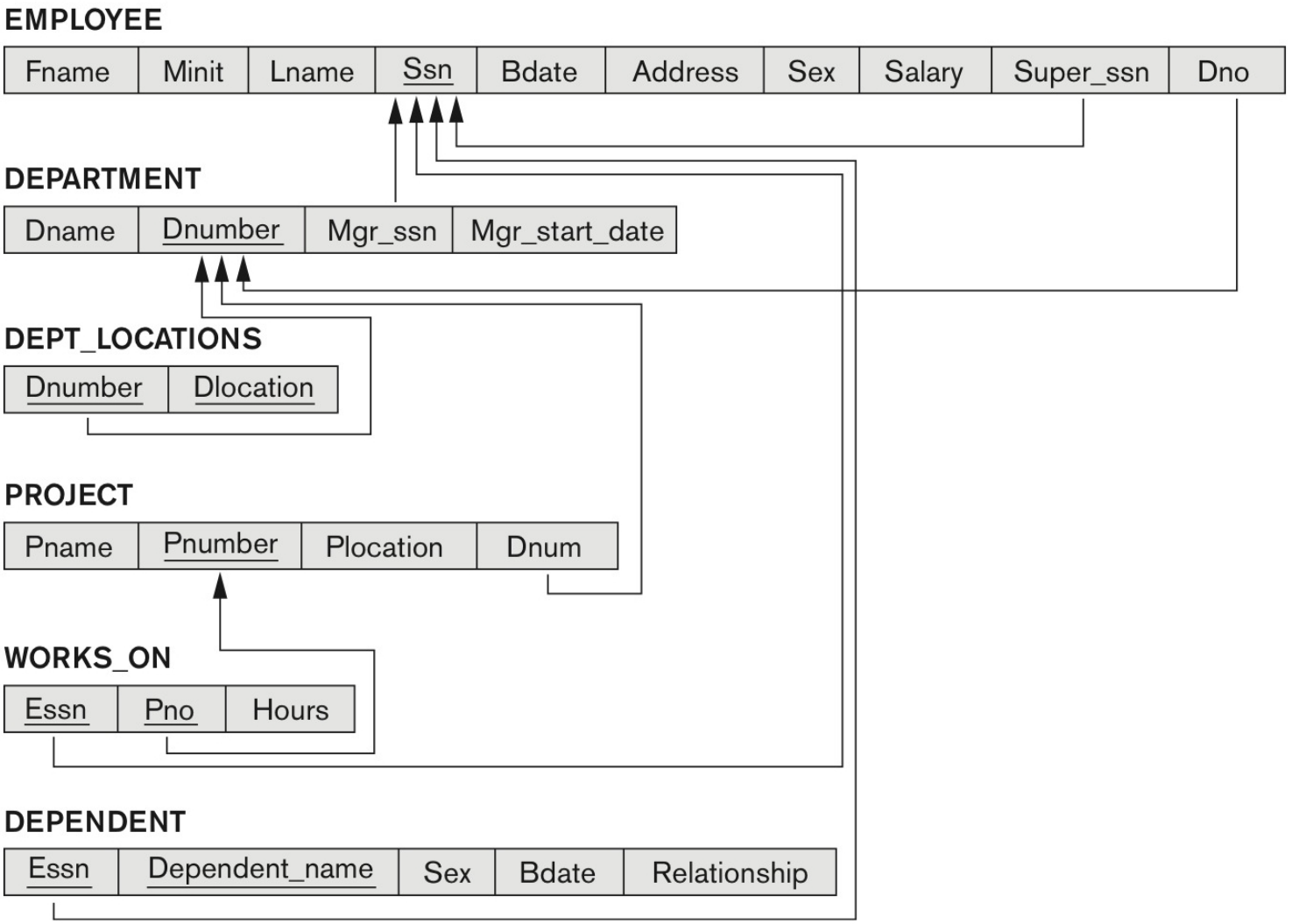


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

The relational model

- A relation is a mathematical concept based on the ideas of sets
- The model was first proposed by Dr. E. F. Codd of IBM Research in 1970 in the following paper:
  - "A Relational Model for Large Shared Data Banks," *Communications of the ACM*, June 1970
- Dr. Codd received the ACM Turing Award

Company relational database



SQL language

- Structured Query Language
- SQL language: considered one of the major reasons for the commercial success of relational databases
- A comprehensive language: Data definition, schema definition, data manipulation, transaction control, indexing, security specification active databases, …
- Variations in existing RDBMS systems
- Base relation and virtual relations (views)

Overview of the DB design process

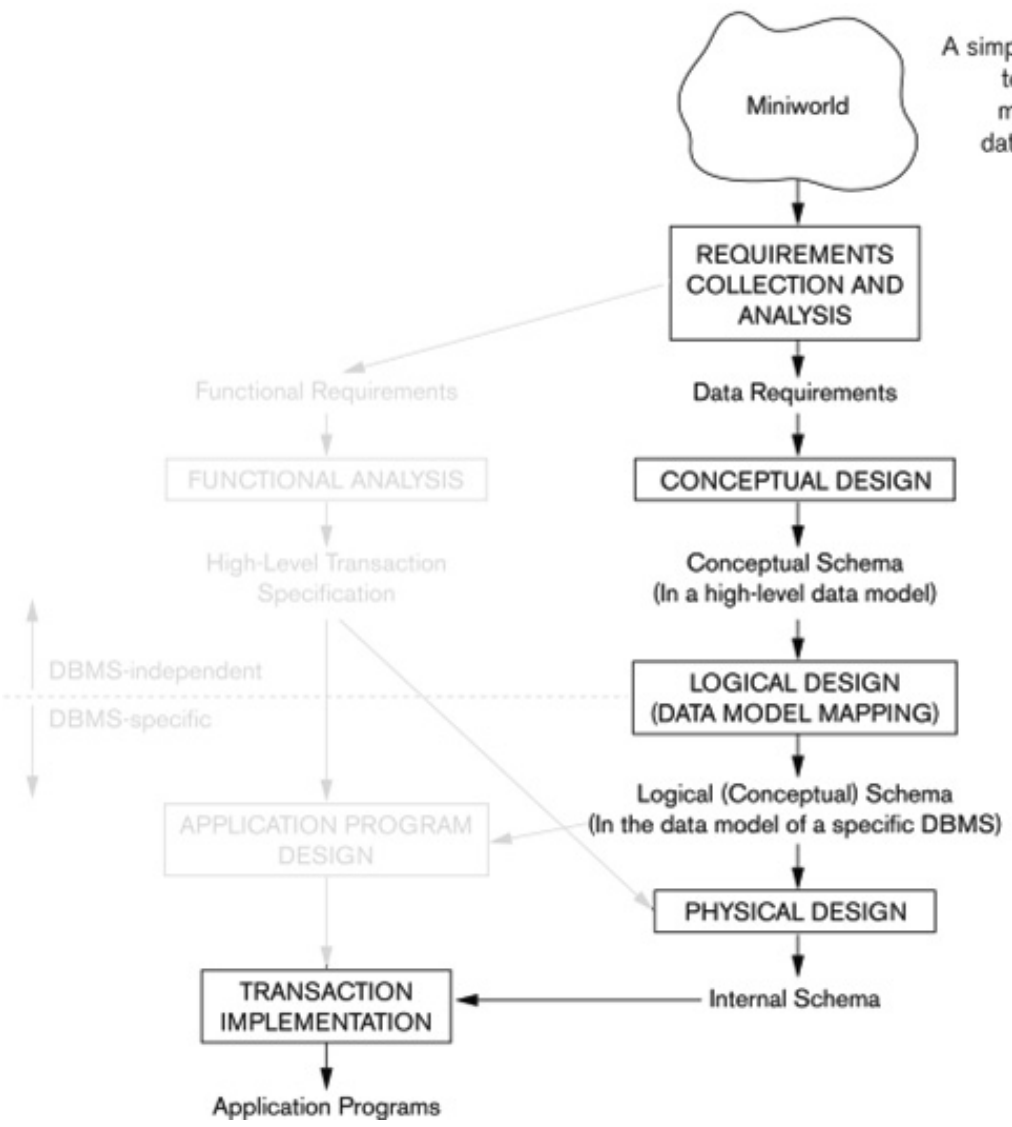


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

Company relational database

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

Company relational database

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

An example of CREATE statement

```
CREATE TABLE PROJECT
        ( Pname                 VARCHAR(15)             NOT NULL,
          Pnumber               INT                     NOT NULL,
          Plocation             VARCHAR(15),
          Dnum                  INT                     NOT NULL,
        PRIMARY KEY (Pnumber),
        UNIQUE (Pname),
        FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber) );
```

- Basic data types
  - Numeric: **INTEGER, INT, REAL, FLOAT**
  - Character string (fixed length): **CHAR (n)**
  - Varying length: **VARCHAR (n)**
  - **BOOLEN**
  - **DATE**
  - ...

Insert data into the tables

- **INSERT** inserts a tuple (row) in a relation (table)
- Attribute values should be listed in the same order as were specified in the **CREATE TABLE** command
- Examples

```
INSERT INTO   EMPLOYEE
VALUES        ( 'Richard', 'K', 'Marini', '653298653', '1962-12-30', '98
                Oak Forest, Katy, TX', 'M', 37000, '65329865 3', 4 );
```

```
INSERT INTO   WORKS_ON_INFO ( Emp_name, Proj_name,
              Hours_per_week )
SELECT        E.Lname, P.Pname, W.Hours
FROM          PROJECT P, WORKS_ON W, EMPLOYEE E
WHERE         P.Pnumber=W.Pno AND W.Essn=E.Ssn;
```

Other data/table commands

- Update: Used to modify attribute values of one or more selected tuples

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

- Delete: Removes tuples from a relation

```
DELETE FROM     EMPLOYEE
WHERE           Lname='Brown';
```

SQL data manipulation

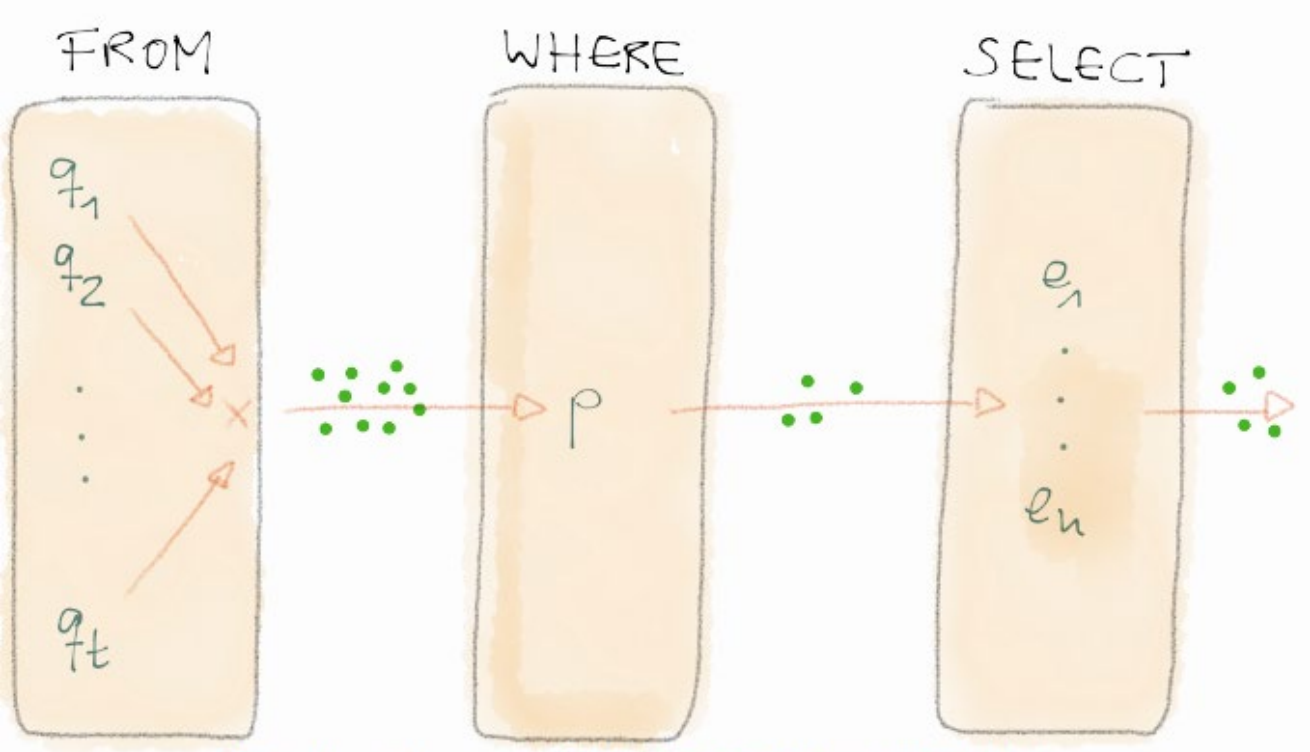
- The ubiquitous **SELECT** construct forms the core of the SQL DML query language
- **SELECT** embodies the principal data language operations
  - Iteration over rows of (multiple) tables, filtering based on predicates
  - Computation over column values (expression evaluation), construction of literal tables
  - Grouping of rows and aggregation of all (or groups of) values in a column
  - And lots more …

SQL: SELECT

SELECT … 3

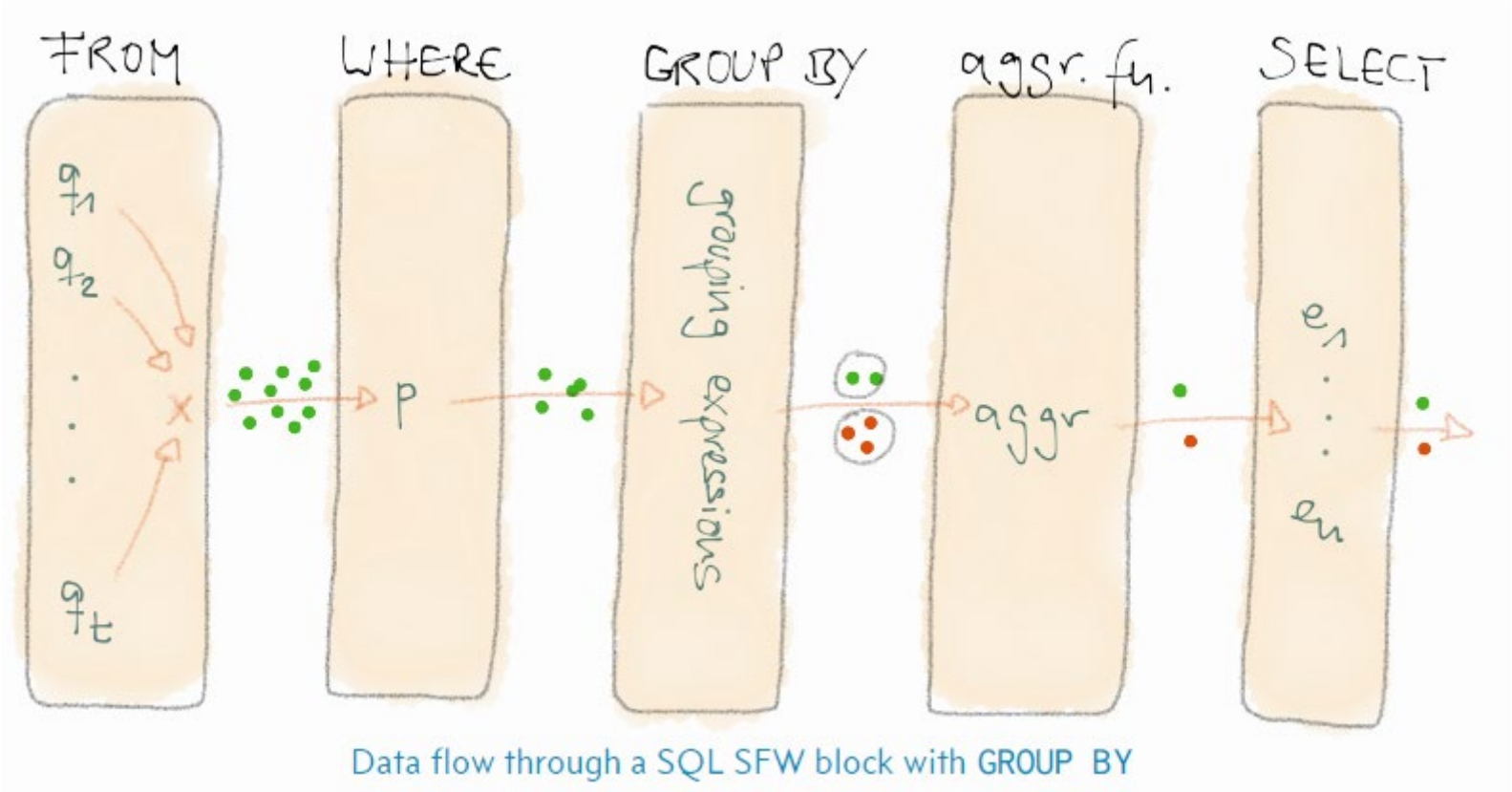
FROM … 1

WHERE … 2



Data flow through a SQL SELECT-FROM-WHERE block

SQL grouping and aggregation



Data flow through a SQL SFW block with GROUP BY

Referencing attributes

- In general, attributes are referenced as R.A where R is a tuple variable and A is an attribute
- When there is no ambiguity, the tuple variable may be deleted

```
SELECT S.lastname F.lastname gpa
FROM Students S, Faculty F
WHERE S.lastname = 'Idena';
```

SQL: grouping and aggregation

- The HAVING and ORDER clauses; very useful in DS queries

```
SELECT Major, AVERAGE(GPA)
FROM STUDENT
WHERE ExpGraduateYr = “2024”
GROUP BY Major
HAVING AVERAGE(GPA) >=3.0
ORDER BY Major;
```

Sample SQL queries

Queries 9 and 10. Select all EMPLOYEE Ssns (Q9) and all combinations of EMPLOYEE Ssn and DEPARTMENT Dname (Q10) in the database.

```
Q9:     SELECT   Ssn
        FROM     EMPLOYEE;

Q10:    SELECT   Ssn, Dname
        FROM     EMPLOYEE, DEPARTMENT;
```

```
Q1C:    SELECT   *
        FROM     EMPLOYEE
        WHERE    Dno=5;

Q1D:    SELECT   *
        FROM     EMPLOYEE, DEPARTMENT
        WHERE    Dname='Research' AND Dno=Dnumber;

Q10A:   SELECT   *
        FROM     EMPLOYEE, DEPARTMENT;
```

Sample SQL queries

Query 4. Make a list of all project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.

```
Q4A:  (SELECT    DISTINCT Pnumber
       FROM      PROJECT, DEPARTMENT, EMPLOYEE
       WHERE     Dnum=Dnumber AND Mgr_ssn=Ssn
                 AND Lname='Smith' )
       UNION
      ( SELECT   DISTINCT Pnumber
       FROM      PROJECT, WORKS_ON, EMPLOYEE
       WHERE     Pnumber=Pno AND Essn=Ssn
                 AND Lname='Smith' );
```

Query 18. Retrieve the names of all employees who do not have supervisors.

```
Q18:  SELECT    Fname, Lname
      FROM      EMPLOYEE
      WHERE     Super_ssn IS NULL;
```

Sample SQL queries

Make a list of all project numbers for projects that involve employee Smith either as worker or as a manager of the department that controls the project

```
Q4A:  SELECT   DISTINCT Pnumber
      FROM     PROJECT
      WHERE    Pnumber IN
               ( SELECT     Pnumber
                 FROM       PROJECT, DEPARTMENT, EMPLOYEE
                 WHERE      Dnum=Dnumber AND
                            Mgr_ssn=Ssn AND Lname='Smith' )
               OR
               Pnumber IN
               ( SELECT     Pno
                 FROM       WORKS_ON, EMPLOYEE
                 WHERE      Essn=Ssn AND Lname='Smith' );
```

Sample SQL queries

Query 16. Retrieve the name of each employee who has a dependent with the same first name and is the same sex as the employee.

```
Q16:  SELECT   E.Fname, E.Lname
      FROM     EMPLOYEE AS E
      WHERE    E.Ssn IN  ( SELECT   Essn
                           FROM     DEPENDENT AS D
                           WHERE    E.Fname=D.Dependent_name
                                    AND E.Sex=D.Sex );
```

List the managers who have at least one dependent

```
SELECT Fname, Lname
FROM Employee
WHERE EXISTS (SELECT *
                FROM DEPENDENT
                WHERE Ssn= Essn)

        AND EXISTS (SELECT   *
                      FROM Department
                      WHERE Ssn= Mgr_Ssn)
```

Summary

- Functions of database systems
  - Persistence
  - Physical and logical data independence
  - High data safety and availability (backup & recovery)
  - Integrity enforcement
  - View management
  - Security via data access control
- SQL: A comprehensive language for relational databases
  - Constructs for data definition, data manipulation, queries, updates, constraint specification, and view definition

Sources

- Elmasri and Navathe, *Fundamentals of Database Systems*, 7th Edition, Pearson, 2016
- Silberschatz, Korth , and Sudarshan, *Database System Concepts*, 7th Edition, McGraw-Hill, 2019
- Prof. Dr. Torsten Grust, Lecture Notes on Foundations of Databases, 2011