CHAPTER 6

Basic Structured Query Language (SQL)

Chapter 6 Outline

- SQL Data Definition and Data Types
- Specifying Constraints in SQL
- Basic Retrieval Queries in SQL
- INSERT, DELETE, and UPDATE Statements in SQL
- Additional Features of SQL

Overview of Database Design Process

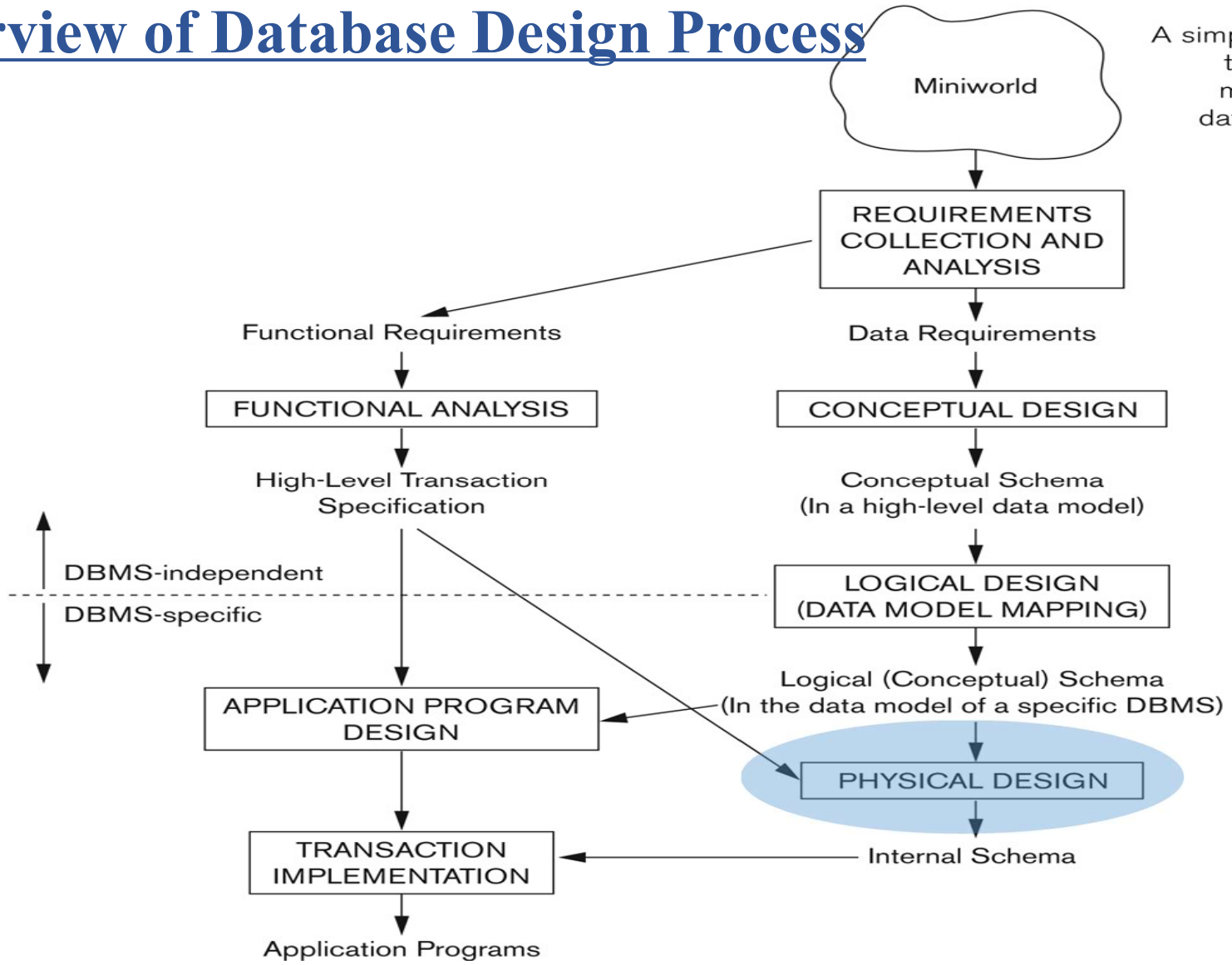
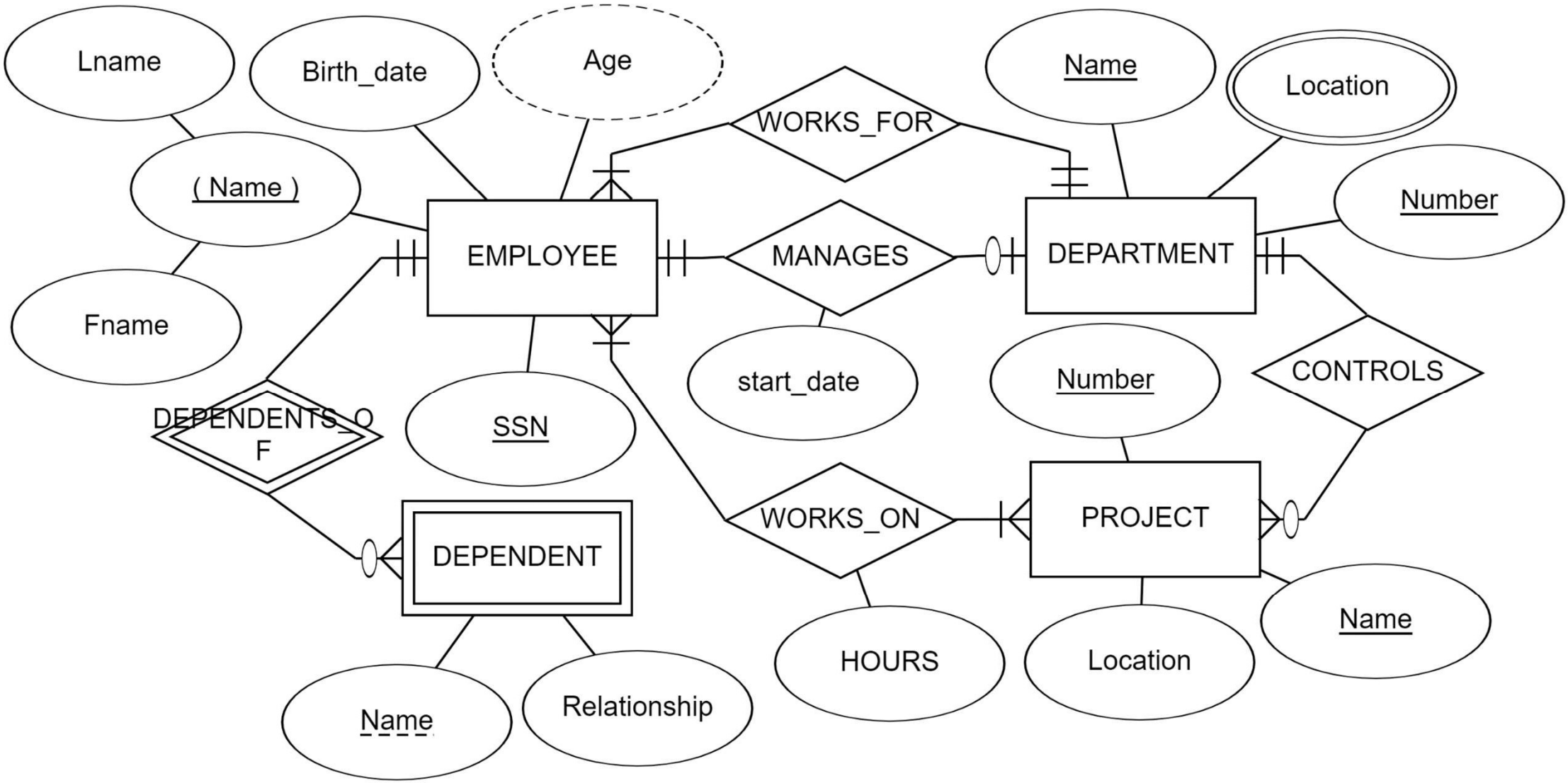
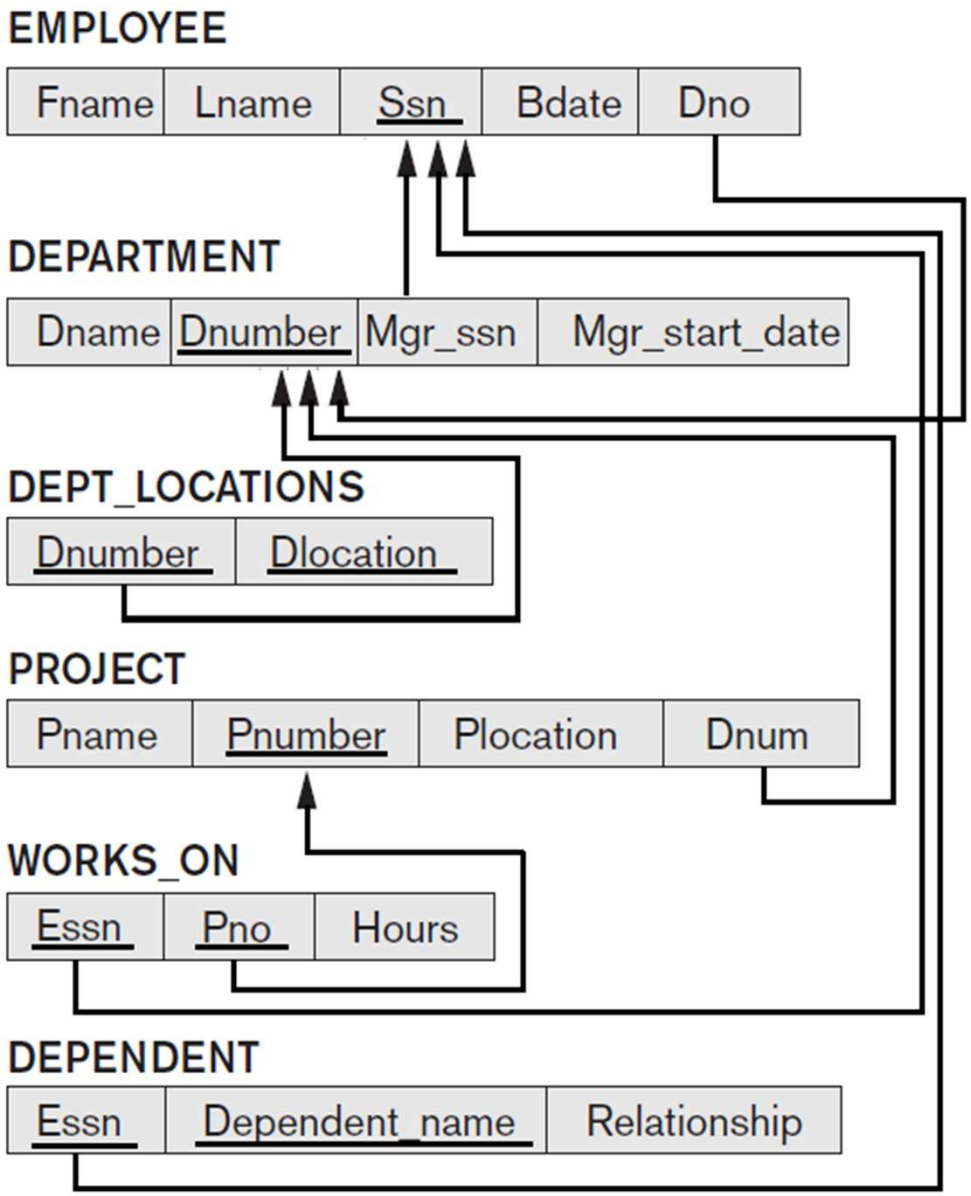


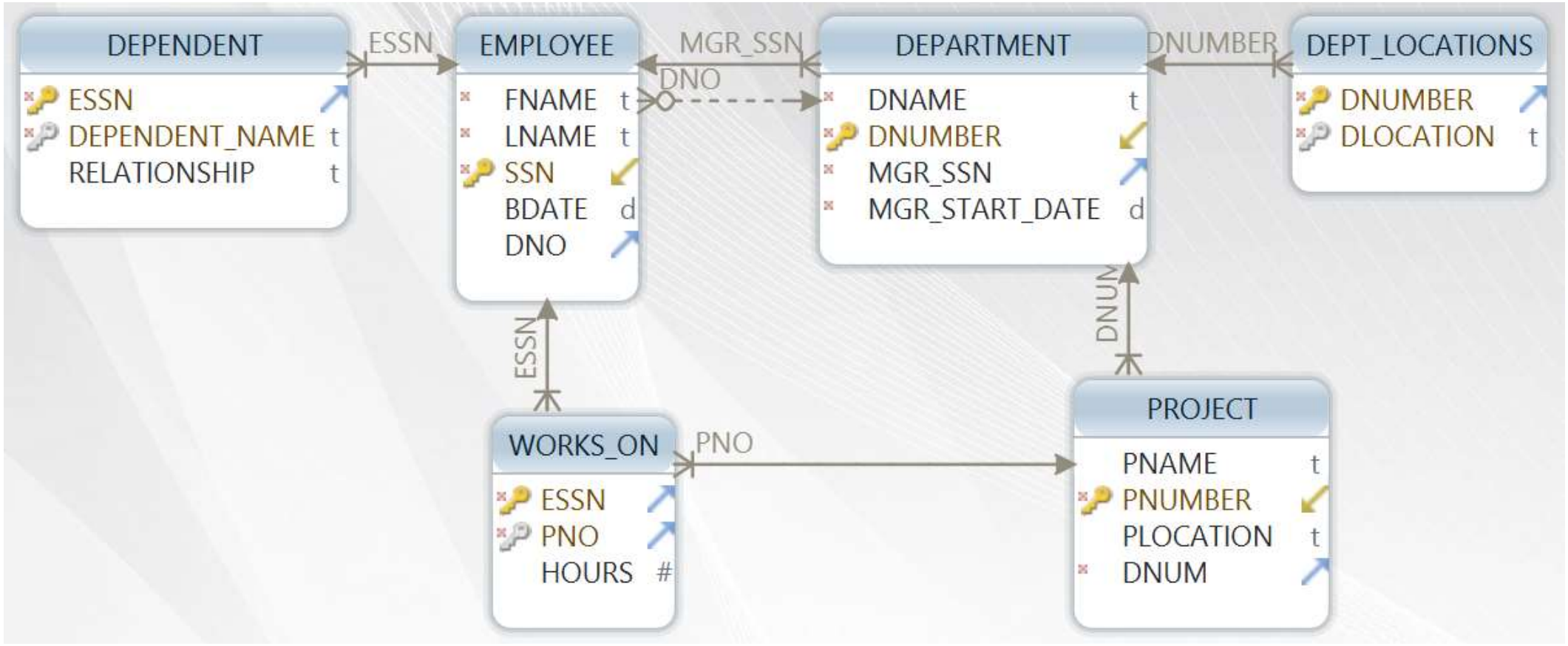
Figure 3.1
A simplified diagram to illustrate the main phases of database design.


Lname
Birth_date
Age
WORKS_FOR
Name
Location
(Name)
EMPLOYEE
MANAGES
DEPARTMENT
Number
Fname
start_date
Number
CONTROLS
DEPENDENTS_OF
SSN
WORKS_ON
PROJECT
DEPENDENT
Name
Relationship
HOURS
Location
Name


EMPLOYEE
Fname
Lname
Ssn
Bdate
Dno
DEPARTMENT
Dname
Dnumber
Mgr_ssn
Mgr_start_date
DEPT_LOCATIONS
Dnumber
Dlocation
PROJECT
Pname
Pnumber
Plocation
Dnum
WORKS_ON
Essn
Pno
Hours
DEPENDENT
Essn
Dependent_name
Relationship

Relational Integrity Constraints

- Constraints are **conditions** that must hold on **all** valid relation states.
- There are three *main types* of (explicit schema-based) constraints that can be expressed in the relational model:
  - **Key** constraints
  - **Entity integrity** constraints
  - **Referential integrity** constraints
- Another schema-based constraint is the **domain** constraint
  - Every value in a tuple must be from the *domain of its attribute* (or it could be **null**, if allowed for that attribute)


DEPENDENT
ESSN
DEPENDENT_NAME t
RELATIONSHIP t
ESSN
EMPLOYEE
FNAME t
LNAME t
SSN
BDATE d
DNO
MGR_SSN
DNO
DEPARTMENT
DNAME t
DNUMBER
MGR_SSN
MGR_START_DATE d
DNUMBER
DEPT_LOCATIONS
DNUMBER
DLOCATION t
ESSN
DNUM
WORKS_ON
ESSN
PNO
HOURS #
PNO
PROJECT
PNAME t
PNUMBER
PLOCATION t
DNUM

Structured Query Language (SQL)

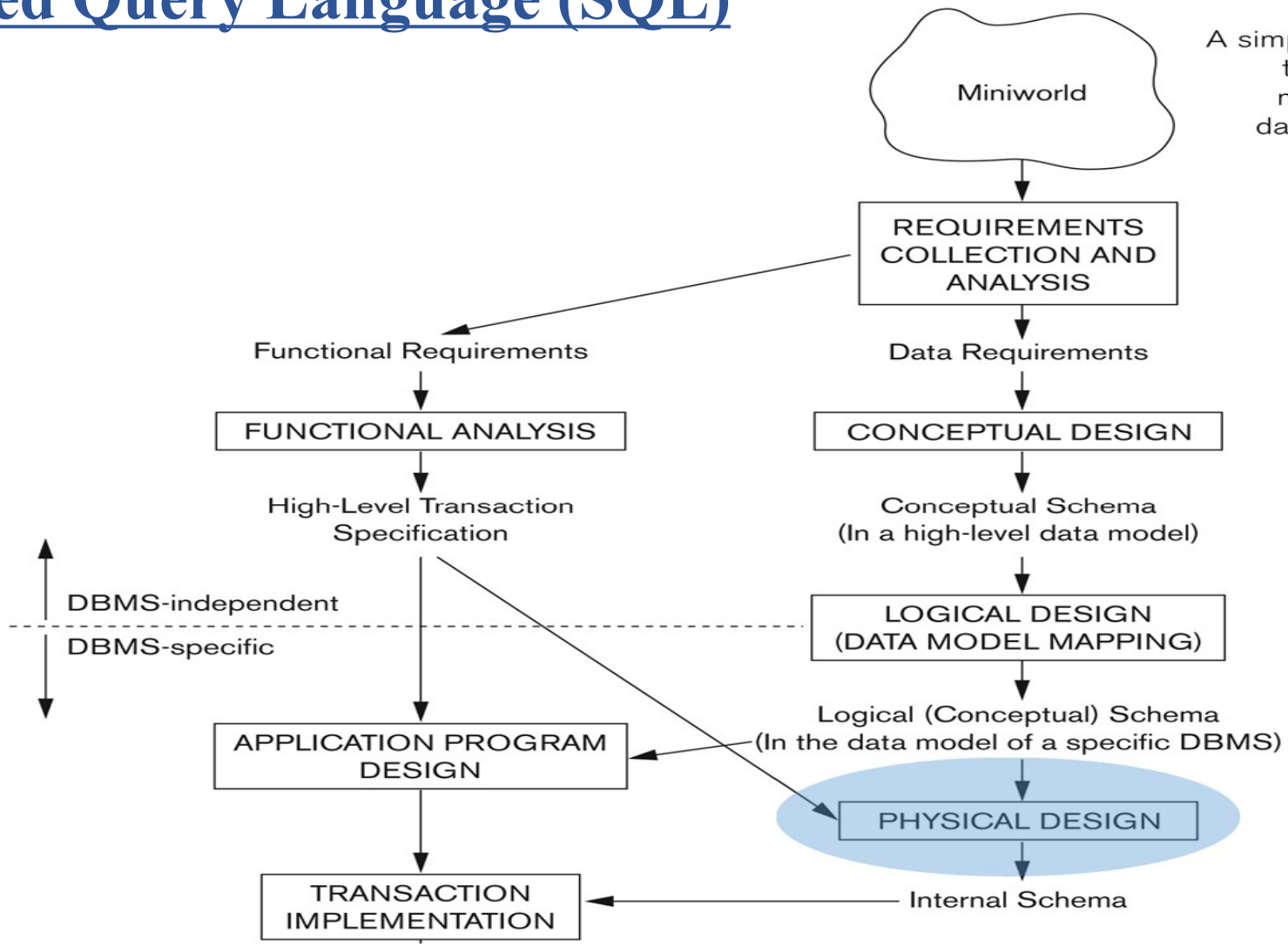
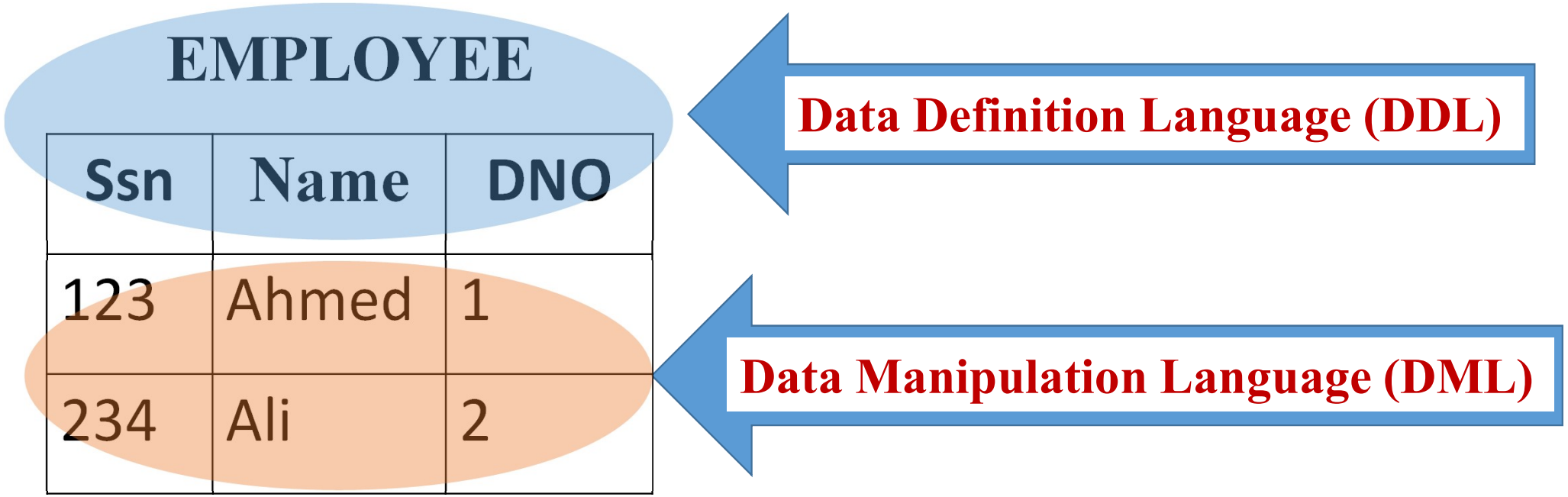


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

data manipulation language (DML)

data definition language (DDL)

Structured Query Language (SQL)



Structured Query Language (SQL)

data definition language (DDL)

Basic SQL

- SQL language
  - Considered one of the major reasons for the commercial success of relational databases
- **SQL**
  - **Structured Query Language**
  - Statements for data definitions, queries, and updates (both DDL and DML)
  - **Core specification**
  - Plus specialized **extensions**

The CREATE TABLE Command in SQL

- Specifying a new relation
  - Provide name of table
  - Specify attributes, their types and initial constraints
- Can optionally specify schema:
  - CREATE TABLE COMPANY.EMPLOYEE ...

    or
  - CREATE TABLE EMPLOYEE ...

An Example of CREATE Statement

EMPLOYEE

Name	Ssn	Salary

```
CREATE   TABLE   EMPLOYEE
(
    Name      VARCHAR(15),
    Ssn       INT,
    Salary    DECIMAL(10,2)
);
```

An Example of CREATE Statement

EMPLOYEE

Name	Ssn	Salary

```
create   table   employee
(
    name      varchar(15),
    ssn       int,
    salary    decimal(10,2)
);
```

EMPLOYEE

NAME	SSN	SALARY

An Example of CREATE Statement

EMPLOYEE

Name	Ssn	Salary

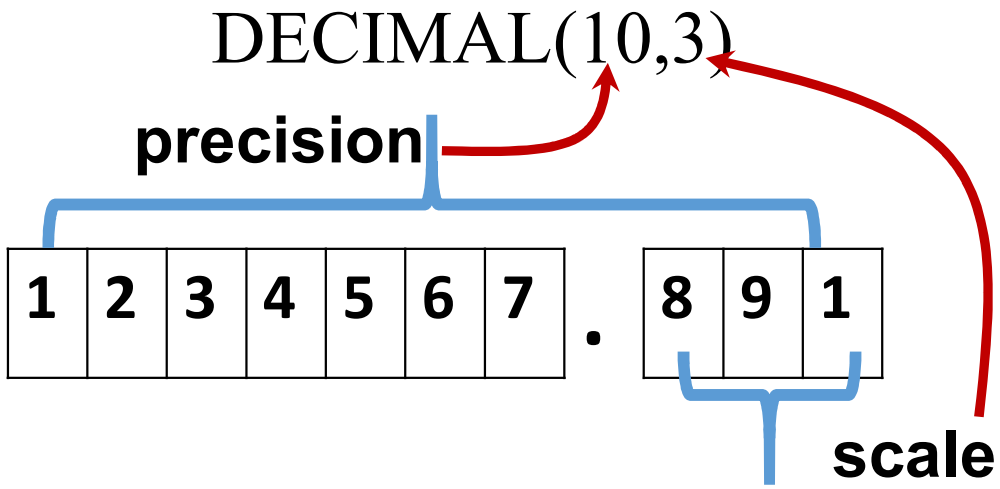
```
Create   Table   Employee
(
    Name      Varchar(15),
    NAME      Varchar(15),
    Ssn       Int,
    Salary    Decimal(10,2)
);
```

Error: Duplicate column name

Attribute Data Types and Domains in SQL

- **Numeric data types**

Numeric data types		
Integer numbers	**INT**	INTEGER and SMALLINT
Floating-point (real) numbers	**FLOAT**	REAL, and DOUBLE PRECISION
	DECIMAL(i, j)	DEC(i, j) or NUMERIC(i, j)



Attribute Data Types and Domains in SQL

- **Numeric data types**

Numeric data types		
Integer numbers	**INT**	INTEGER and SMALLINT
Floating-point (real) numbers	**FLOAT**	REAL, and DOUBLE PRECISION
	DECIMAL(i, j)	DEC(i, j) or NUMERIC(i, j)

```
CREATE  TABLE  EMP
(
    Ssn                 INT,
    Working_hours       FLOAT,
    Salary              DECIMAL(10,3)
);
```

EMP

Ssn	Working_hours	Salary

Attribute Data Types and Domains in SQL

- **Numeric data types**

Numeric data types		
Integer numbers	**INT**	INTEGER and SMALLINT
Floating-point (real) numbers	**FLOAT**	REAL, and DOUBLE PRECISION
	DECIMAL(i, j)	DEC(i, j) or NUMERIC(i, j)

```
CREATE  TABLE  EMP
(
    Ssn              DECIMAL(3,0),
    Working_hours    FLOAT,
    Salary           DECIMAL(10,3)
);
```

EMP

Ssn	Working_hours	Salary

Attribute Data Types and Domains in SQL

- **Character-string data types**

Character-string data types		
Fixed length	**CHAR(n)** **CHAR**	CHARACTER(*n*) for one char as CHAR(1)
Varying length	**VARCHAR(n)**	CHAR VARYING(*n*), CHARACTER VARYING(*n*)

CHAR(5)

' A B C D E '

' A B C D E '

' A B '

Attribute Data Types and Domains in SQL

- **Character-string data types**

Character-string data types		
Fixed length	**CHAR(n)** **CHAR**	CHARACTER(*n*) for one char as CHAR(1)
Varying length	**VARCHAR(n)**	CHAR VARYING(*n*), CHARACTER VARYING(*n*)

CHAR(1) **'A'**

CHAR

Attribute Data Types and Domains in SQL

- **Character-string data types**

Character-string data types		
Fixed length	**CHAR(n)** **CHAR**	CHARACTER(*n*) for one char as CHAR(1)
Varying length	**VARCHAR(n)**	CHAR VARYING(*n*), CHARACTER VARYING(*n*)

```
CREATE  TABLE  EMP
(
     Name       CHAR(30),
     Gender     CHAR,
     Address    VARCHAR(50)
);
```

Attribute Data Types and Domains in SQL

- **DATE** data type

```
CREATE  TABLE  EMP
(
    Name      CHAR(30),
    StartDate          DATE
);
```

NAME	**StartDate**
Ahmed	24-NOV-22
Ali	20-DEC-21
Mohamed	14-JAN-23

Attribute Data Types and Domains in SQL

- **TIMESTAMP** data type

```
CREATE  TABLE  EMP
(
    Name      CHAR(30),
    StartDate          TIMESTAMP
);
```

NAME	StartDate
Ahmed	04-NOV-21 05.56.38.626000

Attribute Data Types and Domains in SQL

Numeric data types		
Integer numbers	**INT**	INTEGER and SMALLINT
Floating-point (real) numbers	**FLOAT**	REAL, and DOUBLE PRECISION
	DECIMAL(i, j)	DEC(i, j) or NUMERIC(i, j)

Character-string data types		
Fixed length	**CHAR(n)** **CHAR**	CHARACTER(*n*) for one char as CHAR(1)
Varying length	**VARCHAR(n)**	CHAR VARYING(*n*), CHARACTER VARYING(*n*)

- **DATE** data type
- **TIMESTAMP** data type

An Example of CREATE Statement

EMPLOYEE

Name	Ssn	Salary	Bdate	Gender

```
CREATE   TABLE   EMPLOYEE
(
    Name      VARCHAR(30),
    Ssn       INT,
    Salary    DECIMAL(10,2),
    Bdate     DATE,
    Gender    CHAR
);
```

An Example of CREATE Statement

DEPARTMENT

Dname	Dnumber	MgrSSN	MgerStartDate

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30),
    Dnumber        DECIMAL(4,0),
    MgrSSN         DECIMAL(10,0),
    MgerStartDate  DATE
);
```

Specifying Attribute Constraints

- Default value of an attribute
  - **DEFAULT** <value>

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30)    DEFAULT  'Research' ,
    Dnumber        DECIMAL(4,0),
    MgrSSN         DECIMAL(10,0)  DEFAULT    1
);
```

Specifying Attribute Constraints

- NULL is not permitted for a particular attribute (NOT NULL)

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30)   NOT   NULL,
    Dnumber        DECIMAL(4,0),
    MgrSSN         DECIMAL(10,0)   NOT   NULL
);
```

Specifying Attribute Constraints

- **PRIMARY KEY** clause
  - Specifies one or more attributes that make up the primary key of a relation

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30),
    Dnumber        DECIMAL(4,0)   PRIMARY   KEY,
    MgrSSN         DECIMAL(10,0)
);
```

Constraint: SYS_C004008

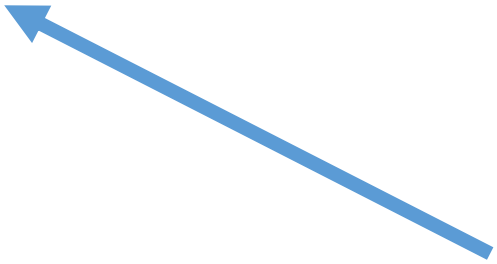
Specifying Attribute Constraints

- **PRIMARY KEY** clause
  - Specifies one or more attributes that make up the primary key of a relation

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30),
    Dnumber        DECIMAL(4,0),
    MgrSSN         DECIMAL(10,0),
    PRIMARY   KEY(Dnumber)
);
```



Constraint: SYS_C004008

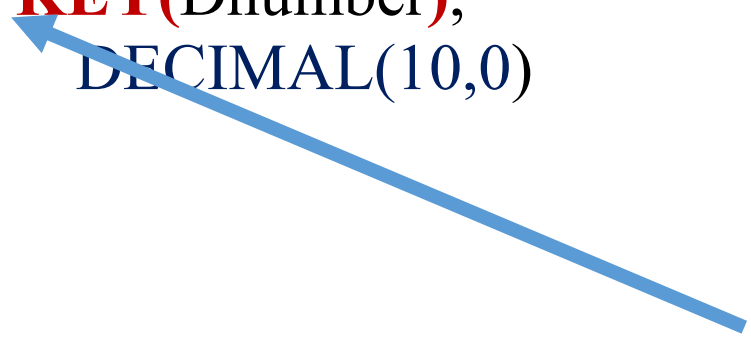
Specifying Attribute Constraints

- **PRIMARY KEY** clause
  - Specifies one or more attributes that make up the primary key of a relation

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30),
    Dnumber        DECIMAL(4,0),
    PRIMARY   KEY(Dnumber),
    MgrSSN         DECIMAL(10,0)
);
```



Constraint: SYS_C004008

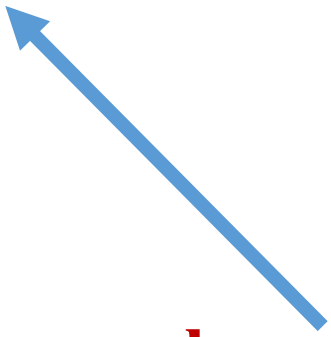
Specifying Attribute Constraints

- **PRIMARY KEY** clause
  - Specifies one or more attributes that make up the primary key of a relation

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30) PRIMARY   KEY,
    Dnumber        DECIMAL(4,0) PRIMARY   KEY,
    MgrSSN         DECIMAL(10,0)
);
```



ERROR: Table can have only one primary key

Specifying Attribute Constraints

- **PRIMARY KEY** clause
  - Specifies one or more attributes that make up the primary key of a relation

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30),
    Dnumber        DECIMAL(4,0),
    MgrSSN         DECIMAL(10,0),
    PRIMARY   KEY (Dname, Dnumber)
);
```

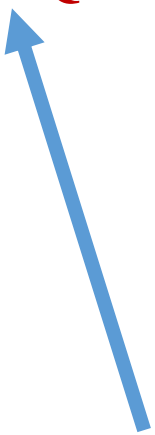
Specifying Attribute Constraints

- **UNIQUE** clause
  - Specifies alternate (secondary) keys (called CANDIDATE keys in the relational model).

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30) UNIQUE,
    Dnumber        DECIMAL(4,0),
    MgrSSN         DECIMAL(10,0)
);
```



Constraint: SYS_C004018

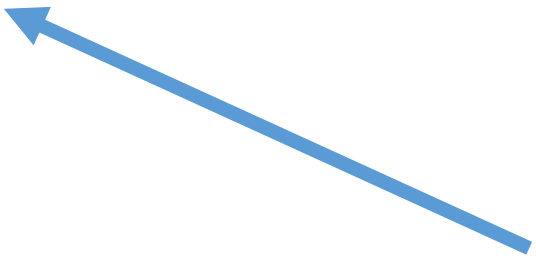
Specifying Attribute Constraints

- **UNIQUE** clause
  - Specifies alternate (secondary) keys (called CANDIDATE keys in the relational model).

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30),
    Dnumber        DECIMAL(4,0),
    MgrSSN         DECIMAL(10,0),
    UNIQUE(Dnumber)
);
```



Constraint: SYS_C004018

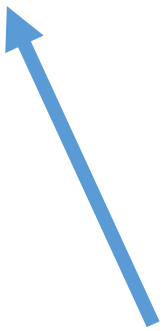
Specifying Attribute Constraints

- **CHECK** clause

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname          VARCHAR(30),
    Dnumber        INT   CHECK ( Dnumber  > 0  AND  Dnumber  <  11),
    MgrSSN         DECIMAL(10,0)
);
```



Constraint: SYS_C004012

Specifying Constraints on Tuples Using CHECK

- **CHECK** clause

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname           VARCHAR(30),
    Dnumber         INT,
    MgrSSN          DECIMAL(10,0),
    CHECK ( Dnumber  > 0  AND  Dnumber  <  11)
);
```

Specifying Constraints on Tuples Using CHECK

- **CHECK** clause

```
CREATE   TABLE   DEPARTMENT
(
    Dname                   VARCHAR(30),
    Dnumber                 INT,
    MgrSSN                  DECIMAL(10,0),
    startDate               Date,
    endDate                 Date,
    CHECK ( startDate <  endDate)
);
```

Specifying Constraints on Tuples Using CHECK

- **CHECK** clause

  CHECK (Dnumber > 0 AND Dnumber < 11)

Logical comparison operators

=, <, <=, >, >=, <>, AND, OR, NOT

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=
BETWEEN	Between a certain range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

Specifying Constraints on Tuples Using CHECK

```
CREATE TABLE DEPARTMENT
(
    DNO         INT                 CHECK(  DNO BETWEEN 50 AND 60  ),
    NAME        VARCHAR(30)  CHECK(  NAME LIKE 'S%'  ),
    MGR_SSN INT                 CHECK(   MGR_SSN  IN (2,5,8,9)   )
);
```

Specifying Constraints on Tuples Using CHECK

```
CREATE TABLE DEPARTMENT
(
    DNO         INT                 CHECK(DNO BETWEEN 50 AND 60),
    NAME        VARCHAR(30)  CHECK(NAME LIKE 'S%'),
    MGR_SSN INT                 CHECK( MGR_SSN  IN (2,5,8,9) )
);
```

Insert new department with the following data (40, 'selling's, 5)

Constraint: SYS_C004002

Constraint: SYS_C004003

Constraint: SYS_C004004

ERROR ORA-02290: check constraint (COMP.SYS_C004002) violated

Giving Names to Constraints

- Using the Keyword **CONSTRAINT**
  - Name a constraint (unique name)
  - Useful for later altering

DEPARTMENT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPARTMENT
(
    Dname           VARCHAR(30),
    Dnumber         INT,
    MgrSSN          DECIMAL(10,0),
    CONSTRAINT  DNO
    CHECK ( Dnumber  > 0  AND  Dnumber  <  11)
);
```

Constraint name: **DNO**

Giving Names to Constraints

- Using the Keyword **CONSTRAINT**
  - Name a constraint (unique name)
  - Useful for later altering

DEPARTMENT

Dname	Dnumber	MgrSSN

DEPT

Dname	Dnumber	MgrSSN

```
CREATE   TABLE   DEPT
(
    Dname          VARCHAR(30),
    Dnumber        INT,
    MgrSSN         DECIMAL(10,0),
    CONSTRAINT  DNO
    CHECK ( Dnumber  > 0  AND  Dnumber  <  11)
);
```

Error: name already used by an existing constraint

Referential Integrity Constraints

- **FOREIGN KEY** clause
  - Default operation: reject update on violation
  - Attach **referential triggered action** clause
    - Options include SET NULL, CASCADE, and SET DEFAULT
    - Action taken by the DBMS for SET NULL or SET DEFAULT is the same for both ON DELETE and ON UPDATE
    - CASCADE option suitable for “relationship” relations

Referential Integrity Constraints

- **FOREIGN KEY** clause

EMPLOYEE

Ssn	Name	DNO

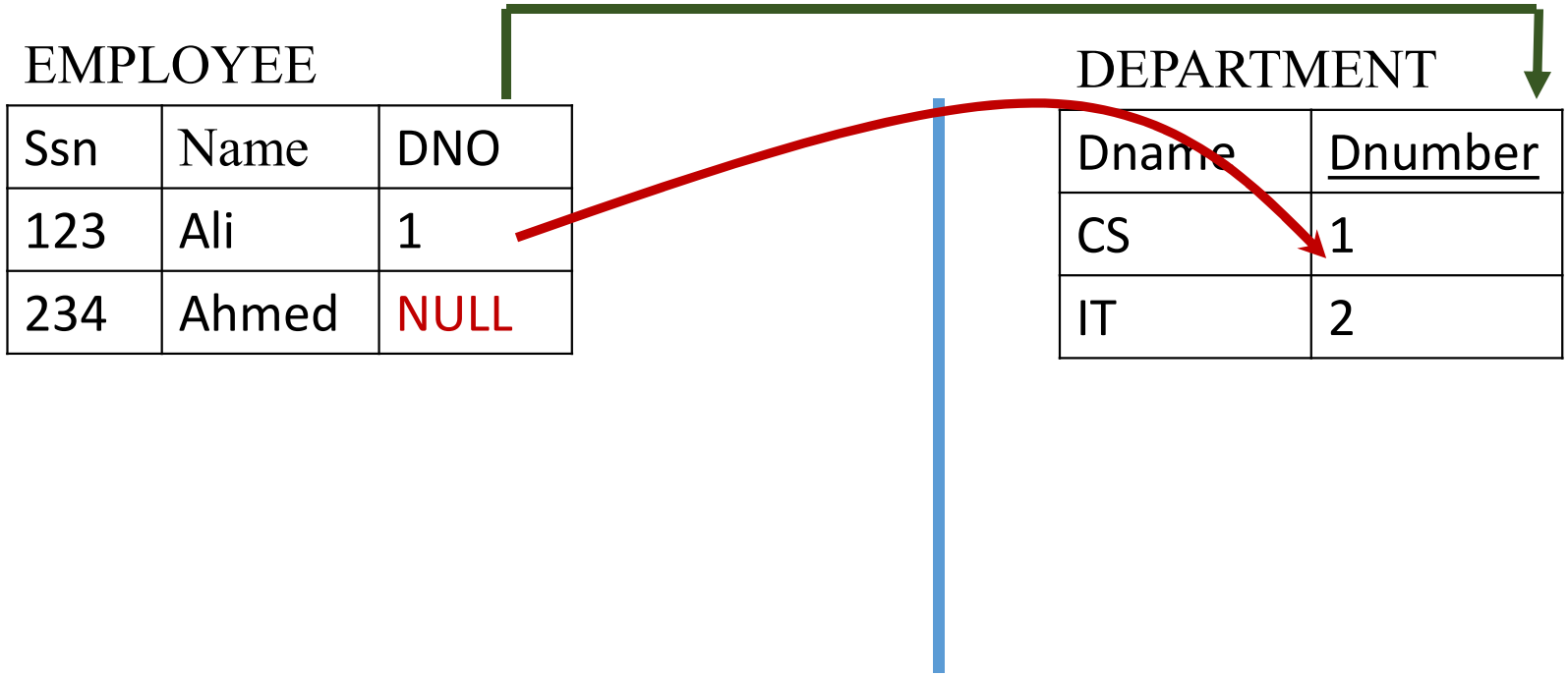
```
CREATE   TABLE   EMPLOYEE
(
    Ssn                 INT,
    Name                VARCHAR(30),
    DNO                 INT,
    CONSTRAINT  FK1
    FOREIGN KEY (DNO) REFERENCES DEPARTMENT(Dnumber)
);
```

Referential Integrity Constraints • **FOREIGN KEY** clause


```
CREATE TABLE EMPLOYEE
(
    Ssn             INT,
    Name            VARCHAR(30),
    DNO             INT,
    CONSTRAINT FK1
    FOREIGN KEY (DNO) REFERENCES DEPARTMENT(Dnumber)
);
```

```
CREATE TABLE DEPARTMENT
(
    Dname     VARCHAR(30),
    Dnumber   INT PRIMARY KEY
);
```

Referential Integrity Constraints • **FOREIGN KEY** clause



Referential Integrity Constraints • **FOREIGN KEY** clause

EMPLOYEE

Ssn	Name	DNO
123	Ali	1
234	Ahmed	NULL

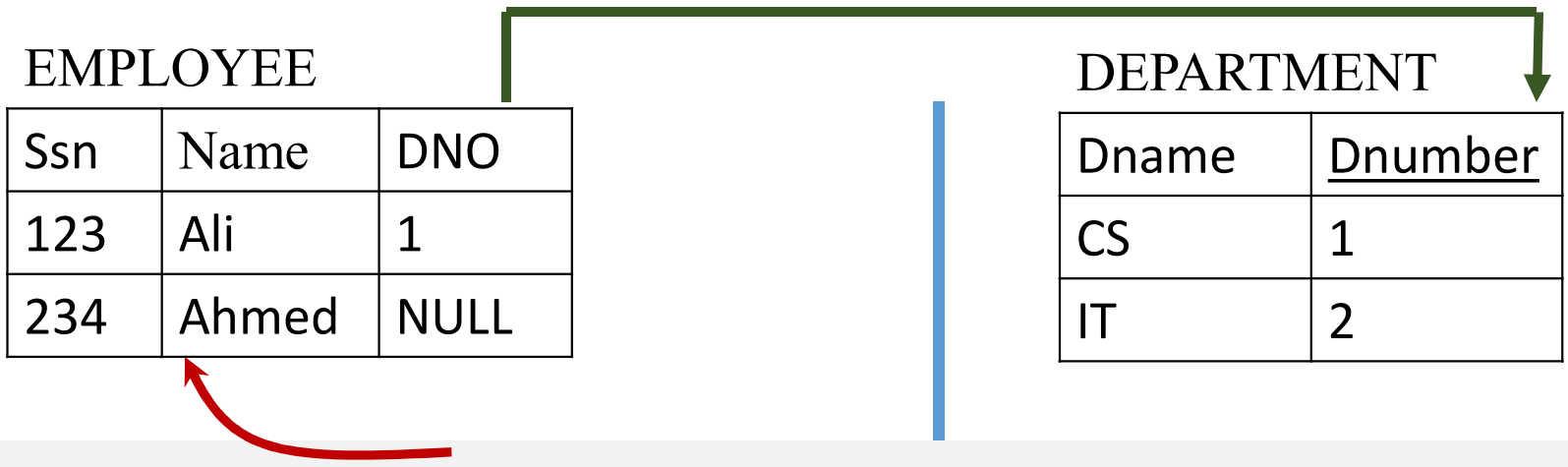
DEPARTMENT

Dname	Dnumber
CS	1
IT	2

```
CREATE   TABLE   EMPLOYEE
(
    Ssn              INT,
    Name             VARCHAR(30),
    DNO              INT,
    CONSTRAINT  FK1
    FOREIGN KEY (DNO) REFERENCES DEPARTMENT(Dnumber)
);
```

```
CREATE   TABLE   DEPARTMENT
(
    . . . . . . . . . . . . . . . . . .
);
```

Referential Integrity Constraints • **FOREIGN KEY** clause



EMPLOYEE

Ssn	Name	DNO
123	Ali	1
234	Ahmed	NULL

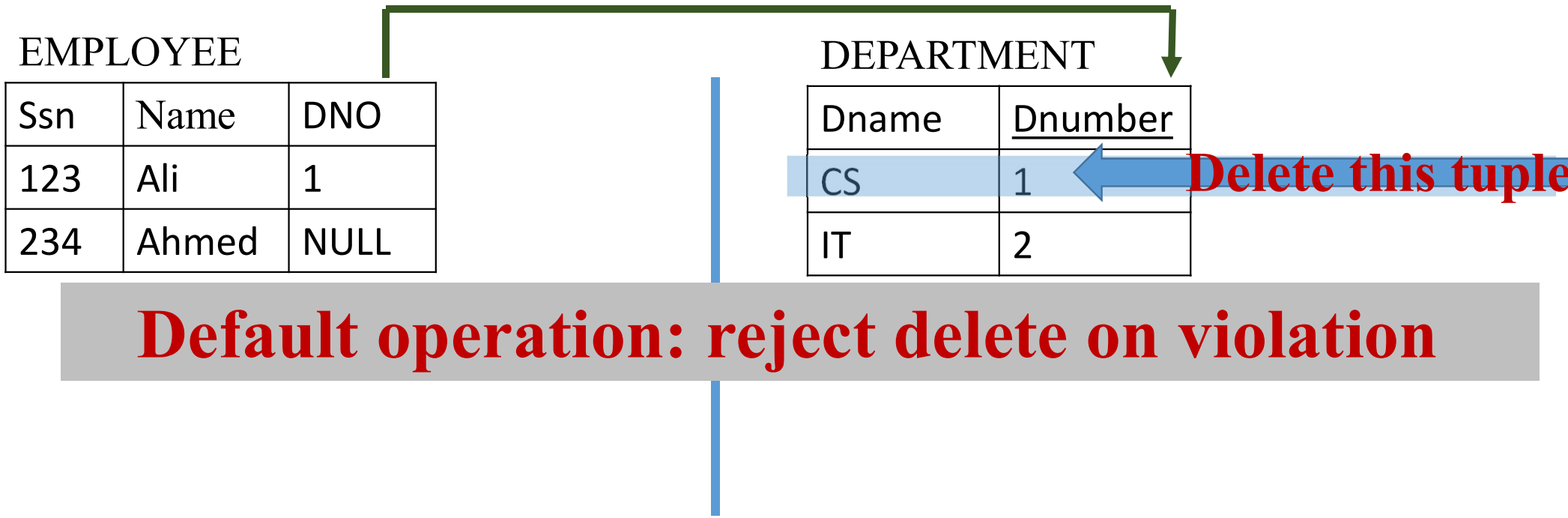
DEPARTMENT

Dname	Dnumber
CS	1
IT	2

ERROR: We can not insert new employee with the following data (345,'Hasan', 3)

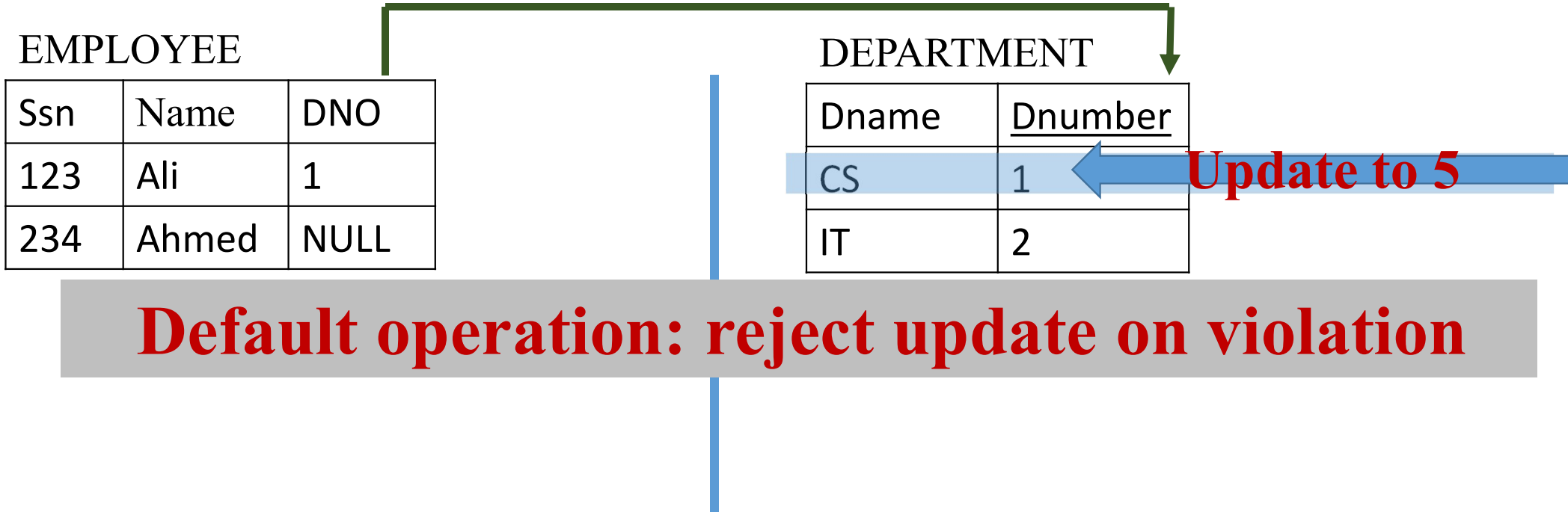
Default operation: reject insert on violation

Referential Integrity Constraints • **FOREIGN KEY** clause



Default operation: reject delete on violation

Referential Integrity Constraints • **FOREIGN KEY** clause



Default operation: reject update on violation

Referential Integrity Constraints • **FOREIGN KEY** clause

EMPLOYEE

Ssn	Name	DNO
123	Ali	1
234	Ahmed	NULL

DEPARTMENT

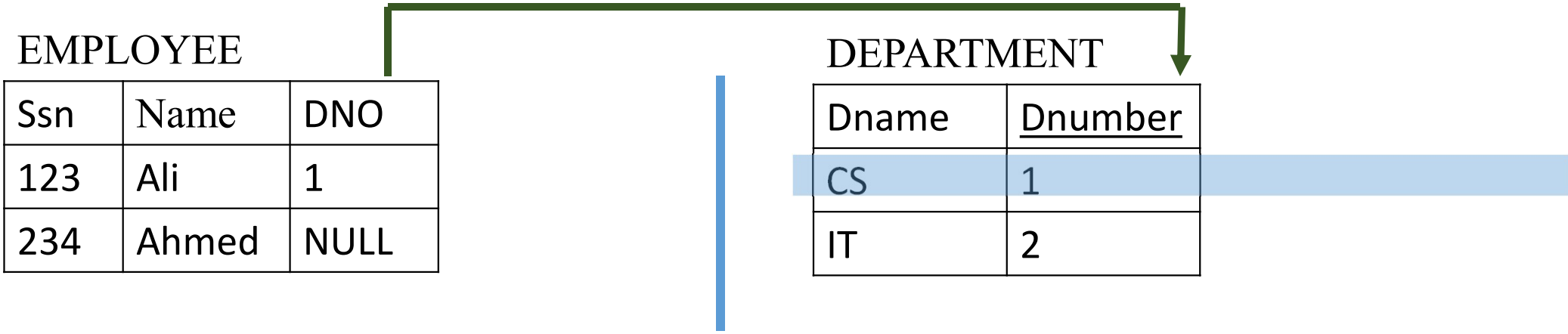
Dname	Dnumber
CS	1
IT	2

FOREIGN KEY (DNO) REFERENCES DEPARTMENT(Dnumber)

ON DELETE
- **SET NULL**
- **SET DEFAULT**
- **CASCADE**

ON UPDATE
- **SET NULL**
- **SET DEFAULT**
- **CASCADE**

Referential Integrity Constraints • **FOREIGN KEY** clause



EMPLOYEE

Ssn	Name	DNO
123	Ali	1
234	Ahmed	NULL

DEPARTMENT

Dname	Dnumber
CS	1
IT	2

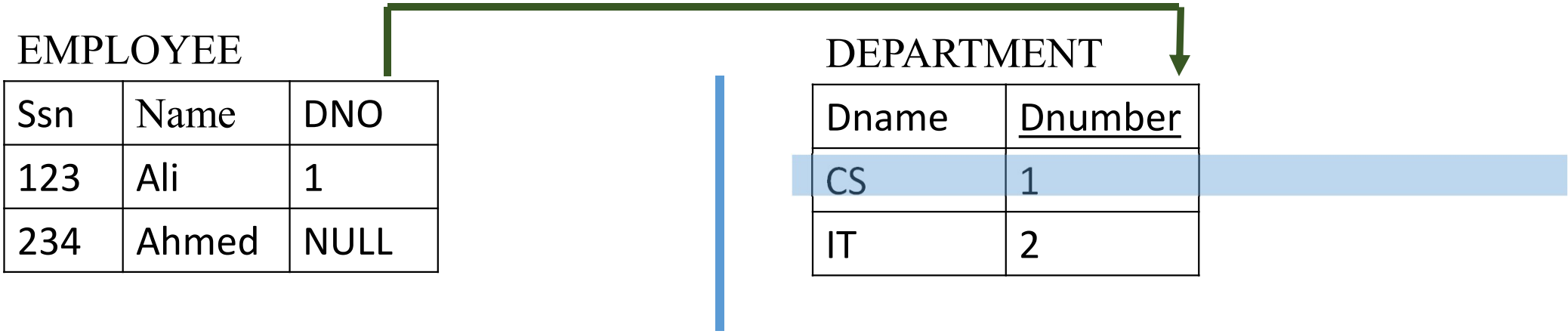
FOREIGN KEY (DNO) REFERENCES DEPARTMENT(Dnumber)

ON DELETE
- **SET NULL**
- **SET DEFAULT**
- **CASCADE**

ON UPDATE
- **SET NULL**
- **SET DEFAULT**
- **CASCADE**

Not all actions are supported by all DBMS

Referential Integrity Constraints • **FOREIGN KEY** clause



EMPLOYEE

Ssn	Name	DNO
123	Ali	1
234	Ahmed	NULL

DEPARTMENT

Dname	Dnumber
CS	1
IT	2

FOREIGN KEY (DNO) REFERENCES DEPARTMENT(Dnumber)

ON DELETE **SET NULL**
CASCADE

Oracle 10g supports only

Referential Integrity Constraints • **FOREIGN KEY** clause


```
CREATE TABLE EMPLOYEE
(
    Ssn         INT,
    Name        VARCHAR(30),
    DNO         INT,
    CONSTRAINT  FK1
    FOREIGN KEY (DNO) REFERENCES DEPARTMENT(Dnumber)
    ON DELETE CASCADE
);
```

```
CREATE TABLE DEPARTMENT
(
    Dname       VARCHAR(30),
    Dnumber  INT PRIMARY KEY

);
```

The ALTER table command

- **Alter table actions** include:
  - Adding or dropping a column (attribute)
  - Changing a column definition
  - Adding or dropping table constraints

The ALTER table command

DEPARTMENT

Dname	Dnumber

- Adding or dropping a column or constraint

Example:

ALTER TABLE DEPARTMENT DROP COLUMN Dname;

The ALTER table command

DEPARTMENT

Dname	Dnumber

- Adding or dropping a column or constraint

Example:

ALTER TABLE DEPARTMENT DROP CONSTRAINT FK2;

The ALTER table command

DEPARTMENT

Dname	Dnumber

- Adding or dropping a column or constraint

Example:

ALTER TABLE DEPARTMENT ADD MgrSSN INT;

The ALTER table command

DEPARTMENT

Dname	Dnumber

- Adding or dropping a column or constraint

Example:

```
ALTER   TABLE DEPARTMENT ADD
        MgrSSN INT;
```

The ALTER table command

DEPARTMENT

Dname	Dnumber

- Adding or dropping a column or constraint

Example:

```
ALTER   TABLE DEPARTMENT ADD
        MgrSSN INT UNIQUE
                   NOT NULL
                   CHECK(MGRSSN> 2 AND MGRSSN<100);
```

The ALTER table command

DEPARTMENT

Dname	Dnumber

- Adding or dropping a column or constraint

Example:

ALTER TABLE DEPARTMENT ADD
CONSTRAINT FK2 UNIQUE(Dnumber);

The ALTER table command

DEPARTMENT

Dname	Dnumber

- Changing a column definition

Example:

ALTER TABLE DEPARTMENT MODIFY
Dname VARCHAR(50);

```
CREATE TABLE DEPARTMENT
(
    Dname VARCHAR(50),
    Dnumber INT PRIMARY KEY,
);
```

The ALTER table command

DEPARTMENT

Dname	Dnumber

- Changing a column definition

Example:

ALTER TABLE DEPARTMENT MODIFY
Dname VARCHAR(50) UNIQUE NOT NULL;

The ALTER table command

- Changing a column definition

Other DBMS may use different formats such as

- **SQL Server / MS Access:**
  - ALTER TABLE *table_name*
    ALTER COLUMN *column_name datatype*;
- **My SQL / Oracle (prior version 10G):**
  - ALTER TABLE *table_name*
    MODIFY COLUMN *column_name datatype*;
- **Oracle 10G and later:**
  - ALTER TABLE *table_name*
    MODIFY *column_name datatype*;

Referential Integrity Constraints • **FOREIGN KEY** clause


```
CREATE TABLE EMPLOYEE
(
    Ssn       INT PRIMARY KEY,
    Name      VARCHAR(30),
    DNO       INT,
    CONSTRAINT FK1
    FOREIGN KEY (DNO) REFERENCES
    DEPARTMENT(Dnumber) ON DELETE
    CASCADE
);
```

```
CREATE TABLE DEPARTMENT
(
    Dname     VARCHAR(30),
    Dnumber   INT PRIMARY KEY,
    MgrSSN    INT,
    CONSTRAINT FK2
    FOREIGN KEY (MgrSSN) REFERENCES
    EMPLOYEE(Ssn)
);
```


```
CREATE TABLE EMPLOYEE(
        Ssn     INT PRIMARY KEY,
        Name    VARCHAR(30),
        DNO     INT,
        CONSTRAINT FK1
        FOREIGN KEY (DNO) REFERENCES
        DEPARTMENT(Dnumber) ON
        DELETE CASCADE
    );
```

```
CREATE TABLE DEPARTMENT(
        Dname   VARCHAR(30),
        Dnumber     INT PRIMARY KEY,
        MgrSSN      INT
    );
```

```
ALTER TABLE DEPARTMENT ADD CONSTRAINT FK2
    FOREIGN KEY (MgrSSN) REFERENCES EMPLOYEE(Ssn);
```

The SQL DROP TABLE Statement

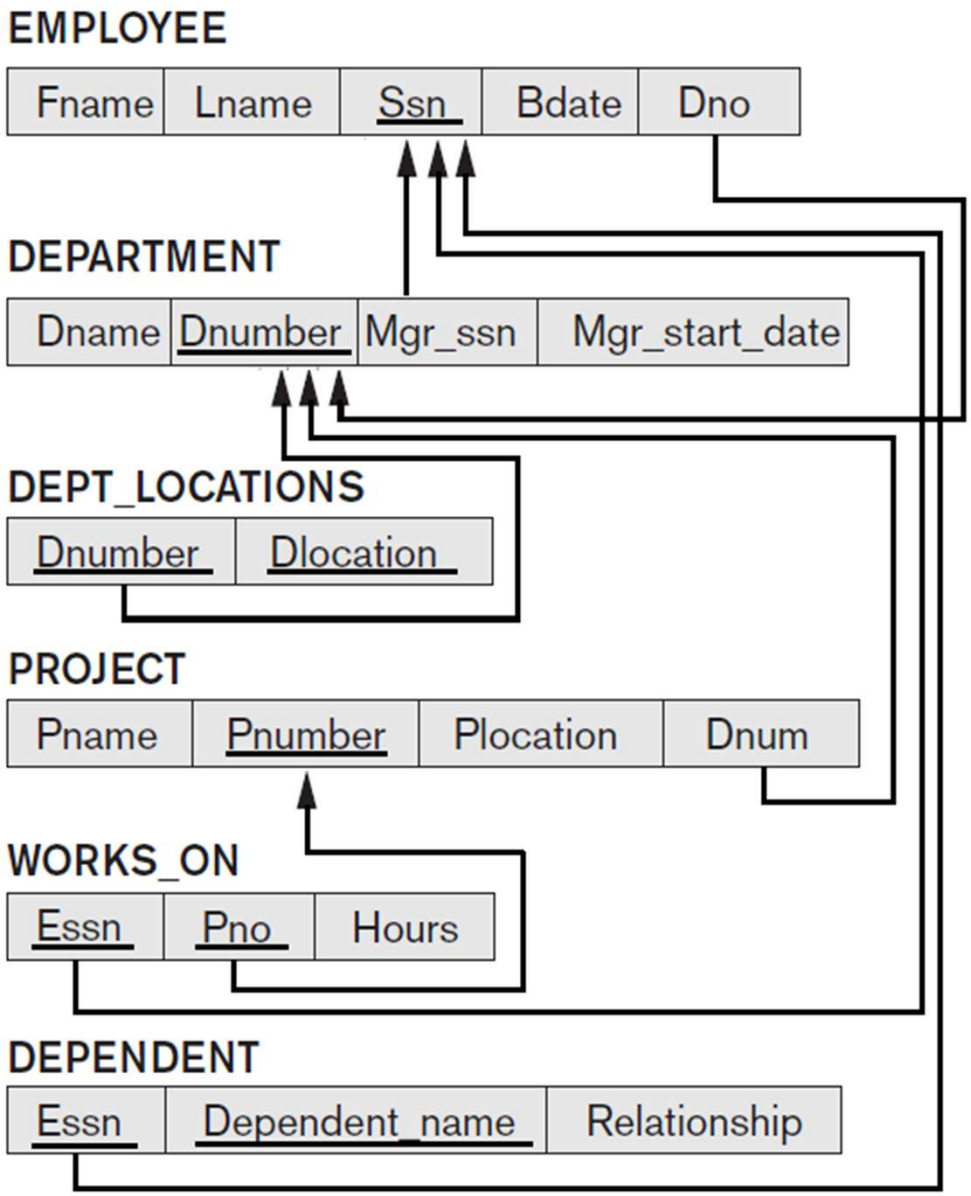
- The DROP TABLE statement is used to drop an existing table in a database.

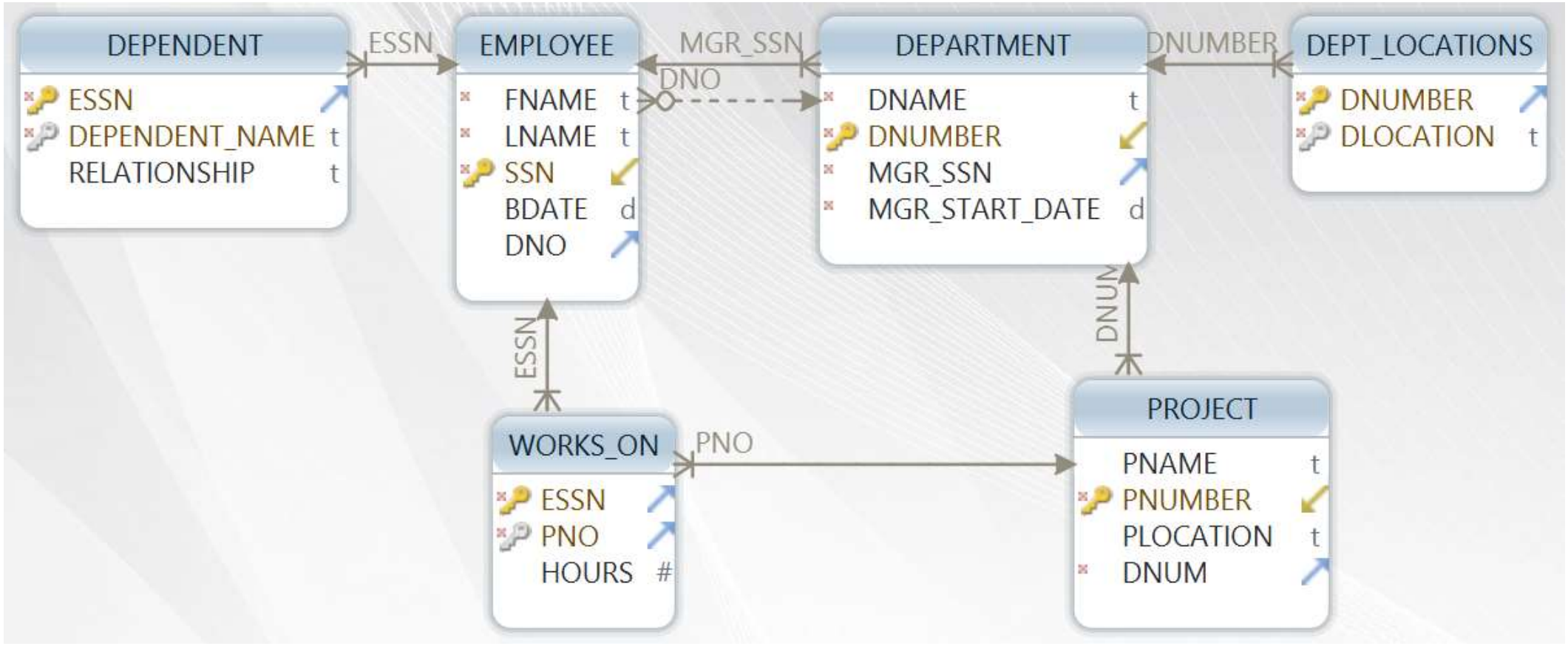
  Syntax

  DROP TABLE EMPLOYEE;

EMPLOYEE

Ssn	Name	DNO


EMPLOYEE
Fname
Lname
Ssn
Bdate
Dno
DEPARTMENT
Dname
Dnumber
Mgr_ssn
Mgr_start_date
DEPT_LOCATIONS
Dnumber
Dlocation
PROJECT
Pname
Pnumber
Plocation
Dnum
WORKS_ON
Essn
Pno
Hours
DEPENDENT
Essn
Dependent_name
Relationship


DEPENDENT
ESSN
DEPENDENT_NAME t
RELATIONSHIP t
ESSN
EMPLOYEE
FNAME t
LNAME t
SSN
BDATE d
DNO
MGR_SSN
DNO
DEPARTMENT
DNAME t
DNUMBER
MGR_SSN
MGR_START_DATE d
DNUMBER
DEPT_LOCATIONS
DNUMBER
DLOCATION t
ESSN
DNUM
WORKS_ON
ESSN
PNO
HOURS #
PNO
PROJECT
PNAME t
PNUMBER
PLOCATION t
DNUM

```
CREATE   TABLE   EMPLOYEE
(
     Fname       VARCHAR(20) NOT NULL,
     Lname       VARCHAR(20) NOT NULL,
     SsnINT PRIMARY KEY,
     Bdate   DATE,
     Dno    INT
);

CREATE   TABLE   DEPARTMENT
(
     Dname       VARCHAR(20) NOT NULL,
     Dnumber INT PRIMARY KEY,
     Mgr_ssn    INT NOT NULL,
     Mgr_start_date   DATE NOT NULL,
     CONSTRAINT DEPT_FK
     FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
);
```

```
ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE_FK
FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber);

CREATE   TABLE   DEPT_LOCATIONS
(
     Dnumber INT,
     Dlocation VARCHAR(20),
     PRIMARY KEY (Dnumber, Dlocation),
     CONSTRAINT LOCATION_FK
     FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)
);
```

```
CREATE   TABLE   PROJECT
(
     Pname VARCHAR(20),
     Pnumber INT PRIMARY KEY,
     Plocation VARCHAR(20),
     Dnum INT NOT NULL,
     CONSTRAINT PROJECT_FK
     FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber)
);
```

```
CREATE   TABLE   WORKS_ON
(
     Essn INT,
     Pno  INT,
     Hours  INT,
     PRIMARY KEY (Essn, Pno),
     CONSTRAINT WORKS_ON_FK1
     FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
     CONSTRAINT WORKS_ON_FK2
     FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber)
);
```

```
CREATE   TABLE   DEPENDENT
(
     Essn INT,
     Dependent_name VARCHAR(20),
     Relationship VARCHAR(20),
     PRIMARY KEY (Essn, Dependent_name),
     CONSTRAINT DEPENDENT_FK1
     FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn)
);
```

Summary

Attribute Data Types and Domains in SQL

Numeric data types		
Integer numbers	**INT**	INTEGER and SMALLINT
Floating-point (real) numbers	**FLOAT**	REAL, and DOUBLE PRECISION
	DECIMAL(i, j)	DEC(i, j) or NUMERIC(i, j)
Character-string data types		
Fixed length	**CHAR(n)** **CHAR**	CHARACTER(*n*) for one char as CHAR(1)
Varying length	**VARCHAR(n)**	CHAR VARYING(*n*), CHARACTER VARYING(*n*)

- **DATE** data type
- **TIMESTAMP** data type

Summary

```
CREATE  TABLE  DEPARTMENT
(
    Dname          VARCHAR(30) UNIQUE,
    Dnumber        DECIMAL(4,0) PRIMARY KEY,
    MgrSSN         DECIMAL(10,0) DEFAULT   1,
    MgerStartDate  DATE  NOT  NULL,
    CHECK ( Dnumber  > 0  AND  Dnumber  <  11),
    CONSTRAINT  FK2 FOREIGN KEY (MgrSSN) REFERENCES
    EMPLOYEE(Ssn) ON DELETE CASCADE
);
```

Summary

- ALTER TABLE DEPARTMENT DROP COLUMN Dname;
- ALTER TABLE DEPARTMENT DROP CONSTRAINT FK2;
- ALTER TABLE DEPARTMENT ADD MgrSSN INT;
- ALTER TABLE DEPARTMENT ADD CONSTRAINT FK2 UNIQUE(Dnumber);
- ALTER TABLE DEPARTMENT MODIFY Dname VARCHAR(50);
- DROP TABLE EMPLOYEE;