Database Management Systems

Week 4: Section-A scribe notes

Submitted By

K Laxman	S20170010079
T Nischal	S20170010161
Piyush	S20170010109
M Amulya	S20170010099
Sanjay Kumar	S20170010139

Contents

1.Relational Model Constraints

Relational Model Constraints: Constraints are rules/restrictions imposed on the relational databases to determine which values are allowed in the database and which values are not.

A. Types of Constraints:

Constraints on databases can usually be divided into 3 main categories:

1.Inherent model-based constraints or implicit constraints:

Constraints that are inherent in the data model.These constraints are a part of relational model itself and its characteristics.

Example:

a.Constraint that a relational model doesn't allow duplicate tuples.

b.Constraint that a relational model doesn't allow a list as a value for any attribute.

2.Schema-based or Explicit Constraints:

Constraints that can be directly expressed in schemas of the data model, typically by specifying them in the DDL(Data Definition Language).

Example:

a. maximum cardinality ratio constraint in the ER model.

3.Application based or semantic constraints:

Constraints that cannot be directly expressed in schemas of the data model, and hence must be expressed and enforced by the application programs.

These constraints relate to the meaning as well as behavior of attributes, and are difficult to express and enforce within the data model, so they are usually checked within the application programs that perform database updates.

Example:

“the max. no. of hours per employee for all projects he or she works on is 56 hrs per week.”

Various Schema-based Constraints:

1. Domain Constraints:

Domain Constraints are used to specify set of all possible values that a specific attribute is allowed to take.

Example :

Usa_phone_numbers(domain): The set of ten-digit phone numbers valid in the United States.

2. Key Constraints:

Super Key of Relation (R) :

A SuperKey(SK) Is a set of attributes of R which obey the following condition:

1.No two distinct tuples in any relation state(r) of R can have the same value for SK.

2. That is, for any two distinct tuples t1 and t2 in r(R), t1[SK] != t2[SK].

3. And the above condition must hold in any valid state r(R).

Key of Relation (R) :

A key K of a relation schema R is a superkey of R with the following additional property:

1.Removal of any attribute A from K leaves a set of attributes K' that is not a superkey(does not possess the superkey uniqueness property) of R anymore i.e. It is a 'minimal superkey.'

Note: A Key is also a Superkey but not vice versa.

Example of keys and SuperKeys :

Consider the following STUDENT Relational Schema:

STUDENT(Name, Ssn, Mobile_no, Address, Age, Gpa):

STUDENT has one key:

1. Key1-{Ssn}

Key1 is also a Superkey of Student.

{Name,Ssn} is a SuperKey but not a 'key'.

Candidate Key:

- A proper subset of a super key, which is also a super key is a candidate key.
- All candidate keys are super keys but not vice versa.
- The set of super keys forms the base for selection of candidate keys.

Primary Key:

- In General, a relation has several candidate keys, one is chosen arbitrarily to be the primary key.
    - The primary key attributes are underlined.

- Example: Consider the STUDENT relation schema:

  STUDENT(Name, Ssn, Mobile_no, Address, Age, Gpa)

  We chose "Ssn" as the primary key.

- The primary key is used to uniquely identify each tuple in a relation.
- If there are one or more candidate keys choose one of them as primary key arbitrarily.Generally,we will choose smallest of candidate keys as primary keys.(but it is not always applicable.)

Relational Database Schema:

- A relational database schema S is a set of relation schemas S = {R1, R2, ..., Rm} and a set of integrity constraints IC.
- Here S is the name given to the “whole database schema.”
- R1, R2, ..., Rn are the names of the individual relation schemas within the database S.

Example of Database Schema{Company Database Schema}:

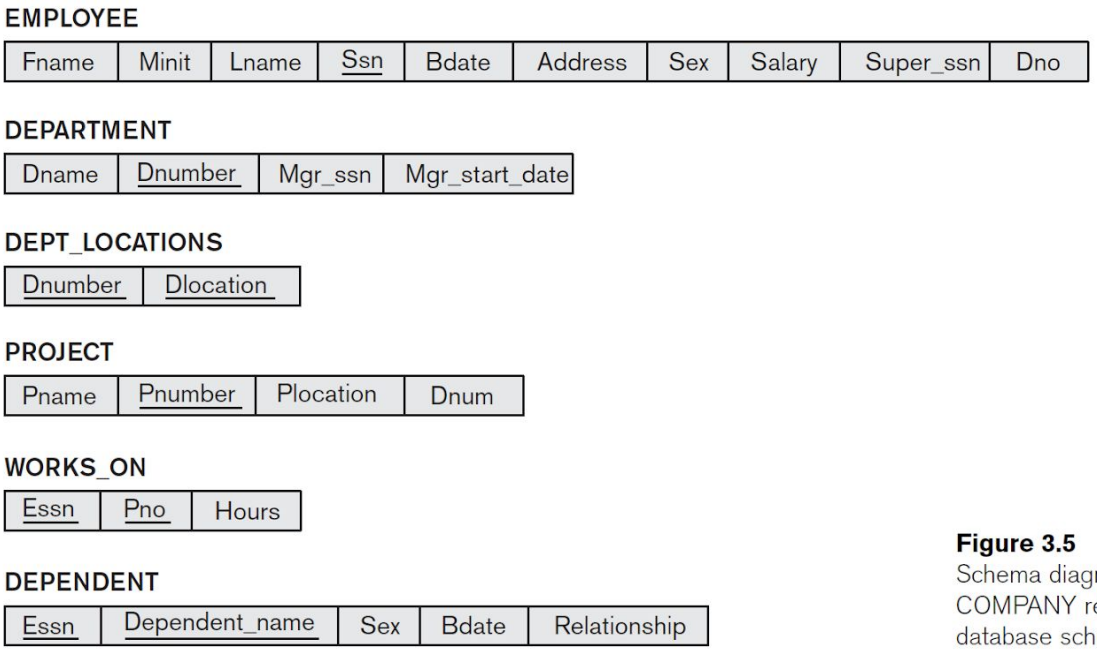


Figure 3.5
Schema diagram for the COMPANY relational database schema.

Relational Database State:

- A relational database state DB of S is a set of relation states DB = {r1, r2, ..., rm} such that each ri is a relation state of Ri and such

that the ri relation states satisfy the integrity constraints specified in IC.

- It is also called relational database snapshot.

Example:

- One possible database state for Company Database is given below:

One possible database state for the COMPANY relational database schema.

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

Entity Integrity Constraint:

- It states Primary Key attribute of any tuple in a relation R cannot have NULL values.
- i.e. t[pk] != NULL for any tuple in r(R).
- This is because the primary key value is used to identify individual tuples in a relation.
- Having NULL values for the primary key implies that we cannot identify some tuples.
- If Primary Key consists of several attributes,then “NULL” value is not allowed in any one of the attributes.

Referential Integrity Constraint(Foreign Key Constraint):

Foreign Key:

A Foreign Key (FK) is a set of attributes in relation R1 is a foreign key of R1 that references relation R2 if it satisfies the following conditions:

- The attributes in FK have the same domain(s) as the primary key attributes PK of R2; the attributes in FK are said to reference or refer to the relation R2.
- A value of FK in a tuple t1 of the current state r1(R1) either occurs as a value of PK for some tuple t2 in the current state r2(R2) or is NULL .
- In the former case, we have t1[FK] = t2[PK], and we say that the tuple t1 references or refers to the tuple t2.

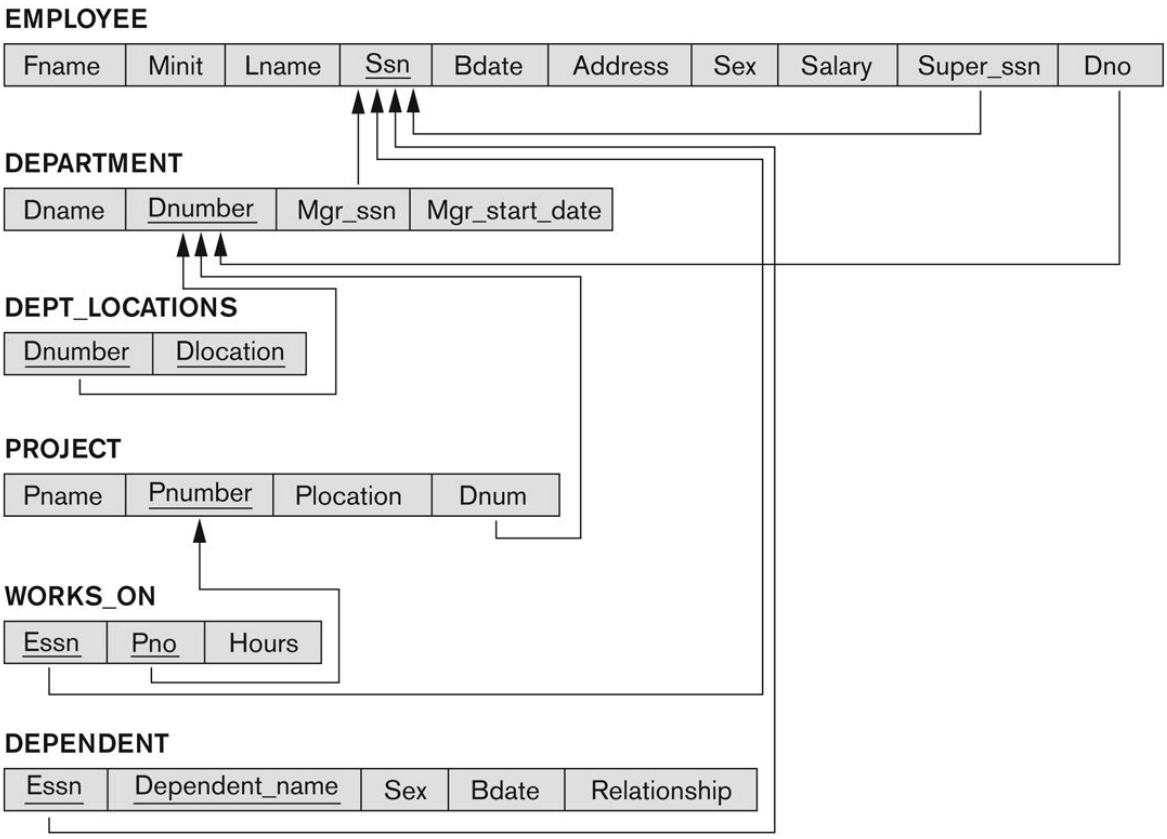
- In this definition, R1 is called the referencing relation and R2 is the referenced relation.
- If these two conditions hold, **a referential integrity** constraint from R1 to R2 is said to hold.
- The **referential integrity constraint** is specified between two relations and is used to maintain the consistency among tuples in the two relations.
- **Note**: A referential integrity constraint can be displayed in a relational database schema as a directed arc from R1.FK to R2.

Displaying a relational database schema and its constraints:

- Each relation schema can be displayed as a row of attribute names
- The name of the relation is written above the attribute names.
- The primary key attribute (or attributes) will be underlined.
- A foreign key (referential integrity) constraints is displayed as a directed arc (arrow) from the foreign key attributes to the referenced table
    - Can also point the the primary key of the referenced relation for clarity.

The next page shows **COMPANY relational schema** diagram with **referential integrity constraints**

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.



2.Dealing with constraints during Update Operations:

List of Update Operations:

- INSERT a tuple.
- DELETE a tuple.
- MODIFY a tuple.

- Integrity constraints should not be violated by the update operations.
- In case of integrity violation, several actions can be taken such as follows :
    - Cancel the operation that causes the violation (RESTRICT or REJECT option).
    - Perform the operation but inform the user of the violation.
    - Trigger additional updates so the violation is corrected (CASCADE option, SET NULL option).
    - Execute a user-specified error-correction routine.

Insert Operation and possible integrity Violations:

- The Insert operation provides a list of attribute values for a new tuple t that is to be inserted into a relation R.
- Insert can violate any four types of integrity constraints:
    - **Domain Constraints:** Domain constraints can be violated if an attribute value is given that does not appear in the corresponding domain or is not of the appropriate data type.
    - **Key Constraints:**Key constraints can be violated if a key value in the new tuple “t” already exists in another tuple in the relation r(R)
    - **Entity integrity:** Entity integrity can be violated if any part of the primary key of the new tuple t is NULL.

- **Referential integrity**:Referential integrity can be violated if the value of any foreign key in t refers to a tuple that does not exist in the referenced relation.

Delete Operation and possible integrity Violations:

- The Delete operation can violate only referential integrity.
- This occurs if primary key value of the tuple being deleted is referenced by foreign keys from other tuples in the database.
- This violation can be countered with several actions:
  - **RESTRICT** option: reject the deletion
  - **CASCADE** option: propagate the new primary key value into the foreign keys of the referencing tuples
  - **SET NULL option**: set the foreign keys of the referencing tuples to NULL.

Modify Operation and possible integrity Violations:

- Updating an attribute that is neither part of a primary key nor of a foreign key usually violates only “domain constraint” or “not null” constraint.
- If the updated attribute is Primary Key it may cause integrity violations **similar to a DELETE followed by an INSERT.**
- We need to specify similar options to **DELETE.**
- Updating a foreign key(FK) may violate referential integrity.

3.Basic SQL

SQL language

Considered one of the major reasons for the commercial success of relational databases

SQL

- The origin of SQL is relational predicate calculus called tuple calculus which was proposed initially as the language SQUARE.
- SQL Actually comes from the word “SEQUEL” which was the original term used in the paper: “SEQUEL TO SQUARE” by Chamberlin and Boyce. IBM could not copyright that term, so they abbreviated to SQL and copyrighted the term SQL.
- Now popularly known as “Structured Query language”.
- SQL is an informal or practical rendering of the relational data model with syntax

Schema and Catalog Concepts in SQL

- **SQL Schema**
    - Identified by schema name

    - Includes Authorization identifier to indicate the user who owns the schema and descriptors for each element

➢ Schema elements include tables, constraints, views, domains, and other constructs (such as authorization grants)

➢ CREATE SCHEMA COMPANY AUTHORIZATION JSMITH;

- **Catalog**
    - A named collection of schemas in an SQL environment.
- SQL environment is basically an installation of an SQL-compliant RDBMS on a computer system.

The CREATE TABLE Command in SQL

➢ used to specify a new relation by giving it a name and specifying its attributes and initial constraints.

➢ Can optionally specify schema

CREATE TABLE COMPANY.EMPLOYEE . . .

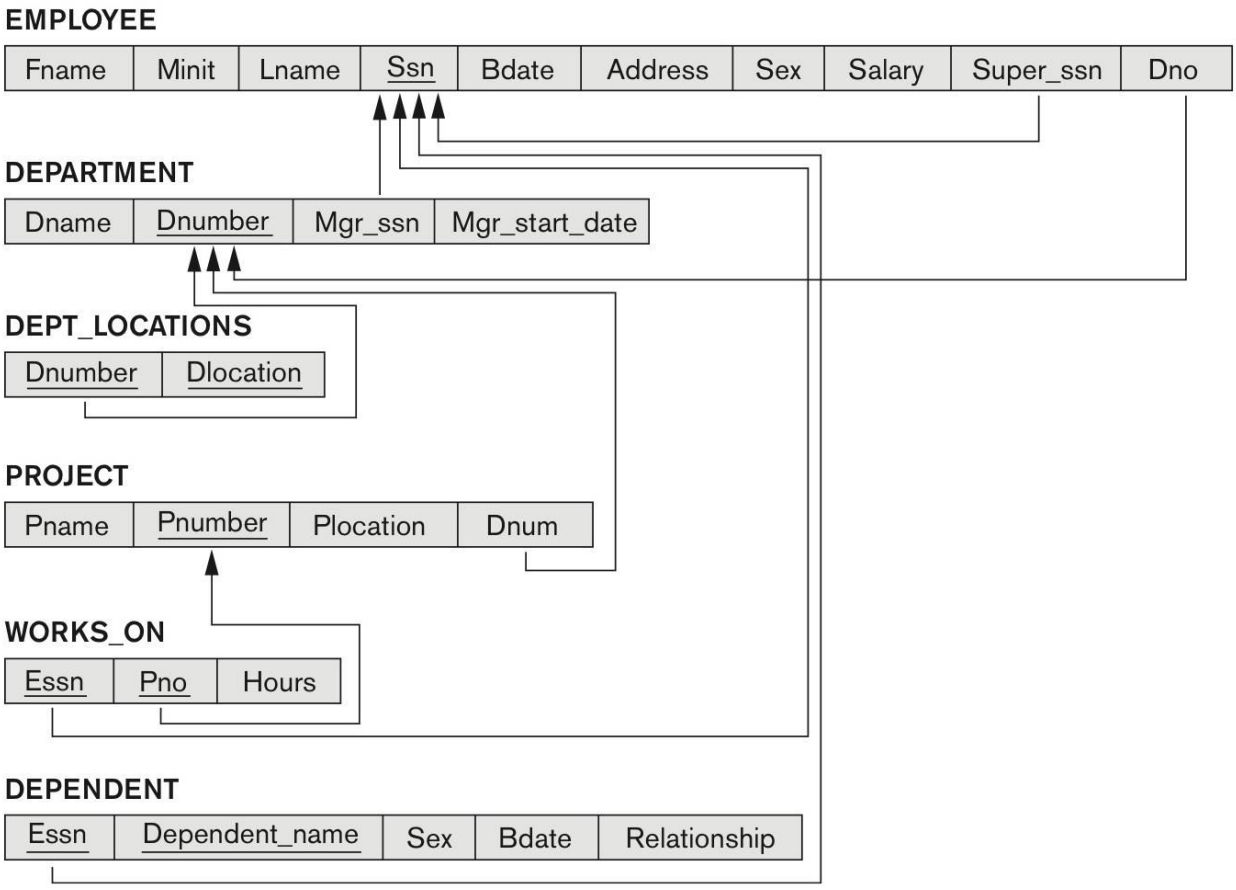
rather than

CREATE TABLE EMPLOYEE . . .

- base tables (or base relations)
    - relations declared through CREATE TABLE statements
    - the relation and its tuples are actually created and stored as a file by the DBMS.
- virtual relations
    - Created through the CREATE VIEW statement
    - may or may not correspond to an actual physical file

COMPANY relational database schema:

(Reference from book Fundamentals of Database System by Elmasri, Navathe)



The sql query for this Company relational database will be :

```
CREATE TABLE EMPLOYEE
        ( Fname                 VARCHAR(15)             NOT NULL,
          Minit                 CHAR,
          Lname                 VARCHAR(15)             NOT NULL,
          Ssn                   CHAR(9)                 NOT NULL,
          Bdate                 DATE,
          Address               VARCHAR(30),
          Sex                   CHAR,
          Salary                DECIMAL(10,2),
          Super_ssn             CHAR(9),
          Dno                   INT                     NOT NULL,
        PRIMARY KEY (Ssn),
CREATE TABLE DEPARTMENT
        ( Dname                 VARCHAR(15)             NOT NULL,
          Dnumber               INT                     NOT NULL,
          Mgr_ssn               CHAR(9)                 NOT NULL,
          Mgr_start_date        DATE,
        PRIMARY KEY (Dnumber),
        UNIQUE (Dname),
        FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn) );
CREATE TABLE DEPT_LOCATIONS
        ( Dnumber               INT                     NOT NULL,
          Dlocation             VARCHAR(15)             NOT NULL,
        PRIMARY KEY (Dnumber, Dlocation),
        FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );
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) );
CREATE TABLE WORKS_ON
        ( Essn                  CHAR(9)                 NOT NULL,
          Pno                   INT                     NOT NULL,
          Hours                 DECIMAL(3,1)            NOT NULL,
        PRIMARY KEY (Essn, Pno),
        FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
        FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber) );
CREATE TABLE DEPENDENT
        ( Essn                  CHAR(9)                 NOT NULL,
          Dependent_name        VARCHAR(15)             NOT NULL,
          Sex                   CHAR,
          Bdate                 DATE,
          Relationship          VARCHAR(8),
        PRIMARY KEY (Essn, Dependent_name),
        FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) );
```

Attribute Data types and Domains in SQL:

Basic data types:

- **Numeric data types:**
  - Integer numbers of various sizes (INTEGER or INT, and SMALLINT)
  - Floating-point (real) numbers: FLOAT or REAL, and DOUBLE PRECISION.
- **Character-string data types:**
  - data types are either fixed length:CHAR(n) or CHARACTER(n), where n is the number of characters or varying length VARCHAR(n) or CHAR VARYING(n) or CHARACTER VARYING(n), where n is the maximum number of characters.
- **Bit-string data types:**
  - data types are either of fixed length n-- BIT(n) or varying length BIT VARYING(n), where n is the maximum number of bits.
- **Boolean data types:**
  - Values of TRUE or FALSE or NULL
- **DATE data type:**
  - It has “10” positions.
  - Components are YEAR, MONTH, and DAY in the form YYYY-MM-DD

    - Multiple mapping functions available in RDBMSs to change date formats
- **Timestamp data type:**
    - Includes the DATE and TIME fields.
    - Plus a minimum of six positions for decimal fractions of seconds
    - Optional WITH TIME ZONE qualifier.
- **Interval data type:**
    - Specifies a relative value that can be used to increment or decrement an absolute value of a date, time, or timestamp
- **Note:**
    - DATE, Timestamp, INTERVAL data types can be castor converted to string formats for comparison.

Specifying Constraints in SQL:

Basic constraints:

Relational Model has 3 basic constraint types that are supported in SQL:

- Key constraint: A primary key value cannot be duplicated
- Entity Integrity Constraint: A primary key value cannot be null

- Referential integrity constraints : The “foreign key “ must have a value that is already present as a primary key, or may be null

NOT NULL Constraint:

- a constraint NOT NULL may be specified if NULL is not permitted for a particular attribute.
    - Example: **Dnumber INT NOT NULL;**
    - Here Dnumber is a attribute of data type INT and NOT NULL constraint has been imposed.

CHECK Constraint:

- Another type of constraint can restrict attribute or domain values using the CHECK clause following an attribute or domain definition
- Example: **Age INT CHECK (Age > 18 AND Age < 99);**

Specifying Key and Referential Integrity Constraints:

- **PRIMARY KEY clause:**
    - The PRIMARY KEY clause specifies one or more attributes that make up the primary key of a relation.

- If a primary key has a single attribute, the clause can follow the attribute directly.
- Example: **Dnumber INT PRIMARY KEY;**
- Example(Multi attribute primary Key):
  - **PRIMARY KEY (Dnumber, Dlocation);**

- **UNIQUE clause:**
  - Specifies alternate (secondary) keys (**CANDIDATE keys** in the relational model).
  - Example**:Dname VARCHAR(15) UNIQUE;**
- **FOREIGN KEY clause:**
  - Referential integrity is specified via the **FOREIGN KEY** clause.
  - a referential integrity constraint can be violated when tuples are inserted or deleted, or when a foreign key or primary key attribute value is modified.
  - **NOTE:**The default action that SQL takes for an integrity violation is to reject the update operation that will cause a violation, which is known as the RESTRICT option.
  - We can 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
    - Example:

      FOREIGN KEY (Dnumber) //foreign key attribute

      REFERENCES DEPARTMENT(Dnumber)

      ON DELETE CASCADE ON UPDATE CASCADE ;

- **<u>Giving Names to Constraints:</u>**
    - a constraint may be given a constraint name, following the keyword CONSTRAINT.
    - The names of all constraints within a particular schema must be unique.
    - A constraint name is used to identify a particular constraint in case the constraint must be dropped later and replaced with another constraint.
    - Example:

CONSTRAINT DEPTPK //constraint name

PRIMARY KEY(Dnumber); // actual constraint.

The page below shows a database schema mentioning various entity integrity ,Referential integrity constraints and referential integrity triggered actions.

```
CREATE TABLE EMPLOYEE
    ( … ,
      Dno          INT          NOT NULL       DEFAULT 1,
    CONSTRAINT EMPPK
      PRIMARY KEY (Ssn),
    CONSTRAINT EMPSUPERFK
      FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
                    ON DELETE SET NULL          ON UPDATE CASCADE,
    CONSTRAINT EMPDEPTFK
      FOREIGN KEY(Dno) REFERENCES DEPARTMENT(Dnumber)
                    ON DELETE SET DEFAULT      ON UPDATE CASCADE);
CREATE TABLE DEPARTMENT
    ( … ,
      Mgr_ssn CHAR(9)          NOT NULL       DEFAULT '888665555',
      … ,
    CONSTRAINT DEPTPK
      PRIMARY KEY(Dnumber),
    CONSTRAINT DEPTSK
      UNIQUE (Dname),
    CONSTRAINT DEPTMGRFK
      FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
                    ON DELETE SET DEFAULT      ON UPDATE CASCADE);
CREATE TABLE DEPT_LOCATIONS
    ( … ,
    PRIMARY KEY (Dnumber, Dlocation),
    FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)
                    ON DELETE CASCADE          ON UPDATE CASCADE);
```

Specifying Constraints on Tuples Using CHECK:

- Additional constraints can be imposed additional CHECK clauses at the end of a CREATE TABLE statement while creating a relational model.
- These can be called **tuple-based constraints** because they apply to **each tuple individually and are checked whenever a tuple is inserted or modified.**
- Example: In Department{ Relational Model } of COMPANY DB schema:
    - CHECK (Dept_create_date<= Mgr_start_date) ;
    - This CHECK clause is executed whenever a tuple is inserted or deleted in Department Table.

Basic Retrieval Queries in SQL:

SQL has one basic statement for retrieving information from a database: the SELECT statement.
SQL allows a table to have two or more tuples that are identical in all their attribute values
Unlike relational model (relational model is strictly set-theory based)
Multiset or bag behavior
Tuple-id may be used as a key

The SELECT-FROM-WHERE Structure of Basic SQL Queries

```
SELECT    <attribute list>
FROM      <table list>
WHERE     <condition>;
```

where

• <attribute list> is a list of attribute names whose values are to be retrieved by the query.
• <table list> is a list of the relation names required to process the query.

• <condition> is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query.

The basic logical comparison operators:
=, <, <=, >, >=, and <>.

The SELECT clause of SQL specifies the projection attributes, and the WHERE clause specifies the selection condition.

Sample Queries and its results:

Query 0. Retrieve the birth date and address of the employee(s) whose name is 'John B. Smith'.

```
Q0:   SELECT   Bdate, Address
      FROM     EMPLOYEE
      WHERE    Fname='John' AND Minit='B' AND Lname='Smith';
```

Output:

Bdate	Address
1965-01-09	731Fondren, Houston, TX

Query 1. Retrieve the name and address of all employees who work for the 'Research' department.

```
Q1:   SELECT   Fname, Lname, Address
      FROM     EMPLOYEE, DEPARTMENT
      WHERE    Dname='Research' AND Dnumber=Dno;
```

Output:

Fname	Lname	Address
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, Houston, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX
Joyce	English	5631 Rice, Houston, TX

Ambiguous Attribute Names, Aliasing, and Tuple Variables

In SQL the same name can be used for two (or more) attributes as long as the attributes are in different relations.

- qualify the attribute name with the relation name to prevent ambiguity
- Example:

```
Q1A:  SELECT   Fname, EMPLOYEE.Name, Address
      FROM     EMPLOYEE, DEPARTMENT
      WHERE    DEPARTMENT.Name='Research' AND
               DEPARTMENT.Dnumber=EMPLOYEE.Dnumber;
```

Aliases or tuple variables :

Declare alternative relation names E and S to refer to the EMPLOYEE relation twice in a query.

Query 8. For each employee, retrieve the employee's first and last name and the first and last name of his or her immediate supervisor.

Answer

SELECT E.Fname, E.Lname, S.Fname, S.Lname

FROM EMPLOYEE ASE, EMPLOYEE ASS

WHERE E.Super_ssn = S.Ssn;