Integrity Constraints

- Part of schema
- Restriction on state (or sequence of states) of data base
- Enforced by DBMS
- Intra-relational - involve only one relation
  - Part of relation schema
  - e.g., all Ids are unique
- Inter-relational - involve several relations
  - Part of relation schema or database schema

16

Kinds of Integrity Constraints

- Static - limitation on state of database
  - Syntactic (structural)
    - e.g., all values in a column must be unique
  - Semantic (involve meaning of attributes)
    - e.g., cannot register for more than 18 credits
- Dynamic - limitation on sequence of database states (supported by some DBMSs, but not in current SQL standard)
  - e.g., cannot raise salary by more than 5%

17

Key Constraint

- Values in a column (or columns) of a relation are unique: at most one row in a relation instance can contain a particular value(s)
- **Key** - set of attributes satisfying key constraint
  - e.g., Id in Student,
  - e.g., (StudId, CrsCode, Semester) in Transcript

18

Key Constraint (con't)

- Minimality - no subset of a key is a key
  - (StudId, CrsCode) is not a key of Transcript
- **Superkey** - set of attributes containing key
  - (Id, Name) is a superkey of Student
- Every relation has a key
- Relation can have several keys:
  - **primary key** (Id in Student) – (cannot be null)
  - **candidate key** ((Name, Address) in Student)

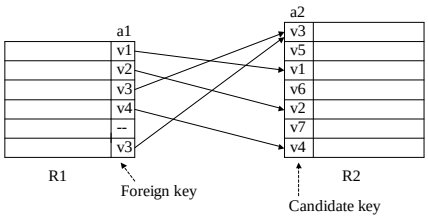
19

Foreign Key Constraint

- R**eferential integrity** => Item named in one relation must correspond to tuple(s) in another that describes the item
  - Transcript (CrsCode) references Course(CrsCode)
  - Professor(DeptId) references Department(DeptId)
- a1 is a **foreign key** of R1 referring to a2 in R2 => if v is a value of a1, there is a *unique* tuple of R2 in which a2 has value v
  - This is a special case of referential integrity: a2 must be a candidate key of R2 (CrsCode is a key of Course)
  - If no row exists in R2 => violation of referential integrity
  - Not all rows of R2 need to be referenced.: relationship is not symmetric (some course might not be taught)
  - Value of a foreign key might not be specified (DeptId column of some professor might be null)

20

Foreign Key Constraint (Example)



21

Foreign Key (con't)

- Names of a1 and a2 need not be the same.
  - With tables:

    Teaching(CrsCode: COURSES, Sem: SEMESTERS, ProfId: INT)
    Professor(Id: INT, Name: STRING, DeptId: DEPTS)

  ProfId attribute of Teaching references Id attribute of Professor
- R1 and R2 need not be distinct.
  - Employee(Id:INT, MgrId:INT, ….)
    - Employee(MgrId) references Employee(Id)
  - Every manager is also an employee and hence has a unique row in Employee

22

Foreign Key (con't)

- Foreign key might consist of several columns
  - (CrsCode, Semester) of Transcript references (CrsCode, Sem) of Teaching
- R1(a1, …an) references R2(b1, …bn)
  - There exists a 1 - 1 relationship between a1,…an and b1,…bn
  - ai and bi have same domains (although not necessarily the same names)
  - b1,…bn is a candidate key of R2

23

Inclusion Dependency

- Referential integrity constraint that is not a foreign key constraint
- Teaching(CrsCode, Semester) references Transcript(CrsCode, Semester) (no empty classes)
  - Reverse relationship *is* a foreign key
- Target attributes are not a candidate key
- No simple enforcement mechanism in SQL

24

Semantic Constraints

- Domain, primary key, and foreign key are examples of structural (syntactic) constraints
- **Semantic constraints** express rules of application:
  - e.g., number of registered students ≤ maximum enrollment

25

SQL

- Language for describing database schema and operations on tables
- **Data Definition Language (DDL):** sublanguage of SQL for describing schema

26

Tables

- SQL entity that corresponds to a relation
- An element of the database schema
- SQL-92 is current standard but will be replaced by SQL:1999
- Database vendors generally deviate from standard

27

Table Declaration

```
CREATE TABLE Student (
  Id: INTEGER,
  Name: CHAR(20),
  Address: CHAR(50),
  Status: CHAR(10))
```

Id	Name	Address	Status
101222333	John	10 Cedar St	Freshman
234567890	Mary	22 Main St	Sophomore

Student

28

Primary/Candidate Keys

```
CREATE TABLE Course (
  CrsCode:CHAR(6),
  CrsName: CHAR(20),
  DeptId: CHAR(4),
  Descr: CHAR(100),
  PRIMARY KEY (CrsCode),
  UNIQUE (DeptId, CrsName))  -- candidate key
```

29

Null

- **Problem**: Not all information might be known when row is inserted (e.g., Grade might be missing from Transcript)
- A column might not be applicable for a particular row (e.g., MaidenName if row describes a male)
- **Solution**: Use place holder - **null**
  - Not a value of any domain (although called null value)
    - Indicates the absence of a value
  - Not allowed in certain situations
    - Primary key, column constrained by NOT NULL

30

Default Value

- Value assigned if attribute value in a row is not specified

```
CREATE TABLE Student (
  Id: INTEGER,
  Name: CHAR(20) NOT NULL,
  Address: CHAR(50),
  Status: CHAR(10) DEFAULT 'freshman',
  PRIMARY KEY (Id))
```

31