CS122 Using Relational Databases and SQL

Introduction to Database Design

Chengyu Sun
California State University, Los Angeles

Video Rental DB

RID	RDate	CID	CName	MID	MName	Price	RQty
1001	05/15/04	201	John	1	Godfather	$4.50	1
1001	05/15/04	201	John	2	Goodfellas	$4.50	1
1001	05/15/04	201	John	3	Casino	$3.50	1
1002	05/23/04	270	Amy	3	Casino	$3.50	1
1003	05/23/04	233	Susan	6	You've Got Mail	$4.50	1
1003	05/23/04	233	Susan	7	Kate & Leopold	$4.50	1
1004	05/24/04	270	Amy	5	Total Recall	$3.50	1

Signs of Bad Design

- Redundancy
- Anomalies
  - Insert
  - Delete
  - Update

Functional Dependency (FD)

- $(A_1, A_2, \ldots, A_n) \rightarrow (B_1, B_2, \ldots, B_m)$
  - If two tuples have the same values of attributes $(A_1, A_2, \ldots, A_n)$, they must have the same values of attributes $(B_1, B_2, \ldots, B_m)$
- Trivial FD
  - $(B_1, B_2, \ldots, B_m) \subseteq (A_1, A_2, \ldots, A_n)$

(RID) → (RDate, CID, CName)
(CID) → (CName)
(MID) → (MName, Price)

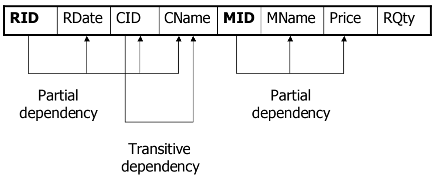
Keys of a Relation

- Key
  - $(A_1, A_2, \ldots, A_n)$ functionally determines *all other* attributes in the relation
  - *Minimal*
- Primary Key
  - Primary key for the Video Rental example??
- Super Key

Partial and Transitive Dependency

- Partial dependency – a non-key column is dependent on part of the primary key
- Transitive dependency – a non-key column is dependent on another non-key column

Dependency Diagram



RID	RDate	CID	CName	**MID**	MName	Price	RQty

Partial dependency

Partial dependency

Transitive dependency

First Normal Form (1NF)

- Primary key defined
- No duplicate columns or multi-valued columns

RID	MID1	MID2	MID3
1001	1	2	3

RID	MID
1001	{1,2,3}

Second Normal Form (2NF)

- 1NF
- No partial dependencies

1NF to 2NF

- Separate each partial dependency into a new table
  - Primary key
  - Column

Third Normal Form (3NF)

- 2NF
- No transitive dependencies

2NF to 3NF

- Separate each transitive dependency into a new table
- And ??

Boyce-Codd Normal Form (BCNF)

- Whenever there's a non-trivial FD $(A_1,A_2,\ldots,A_n) \rightarrow (B_1,B_2,\ldots,B_m)$ in R, $(A_1,A_2,\ldots,A_n)$ is a super key of R

Player	**Instrument**	Teacher
Chloe	Clarinet	Fred
Beryl	Flute	David
Kaitlyn	Drums	Christina
Chloe	Flute	David
Lindsey	Flute	Michele

3NF but not BCNF

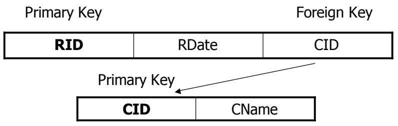
Indexes and Constraints

- Indexes – speed up queries
  - B-tree, R-tree, Quad-tree, Hash, Bitmap …
- Constraints – avoid human errors and speed up queries
  - NOT NULL
  - PRIMARY KEY and UNIQUE
  - FOREIGN KEY and referential integrity

Primary Key And Unique

- PRIMARY KEY
  - One or more fields
  - Implies NOT NULL
- UNIQUE
  - One or more fields
  - Does not imply NOT NULL
  - Can have more than one UNIQUE contraints

Foreign Key and Referential Integrity



- Referenced attribute(s) must be primary key or unique
- Referential integrity
  - A foreign key value is either NULL, or it must exists in the referenced table as a primary or unique key value

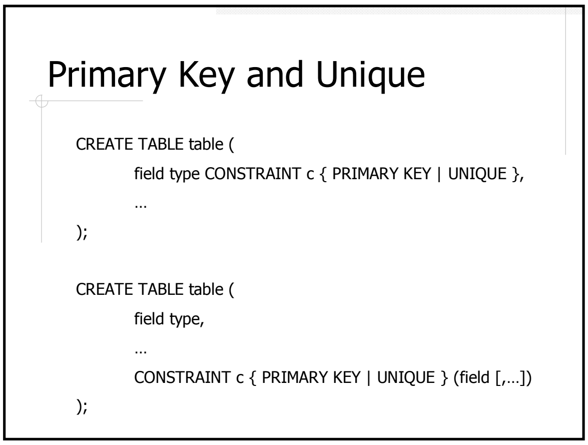
Create and Delete Indexes

```
CREATE [UNIQUE] INDEX index ON table (field [, …])
   [WITH {DISALLOW NULL | IGNORE NULL | PRIMARY}];

DROP INDEX index ON table;
```

- UNIQUE
- DISALLOW NULL
- IGNORE NULL
- PRIMARY

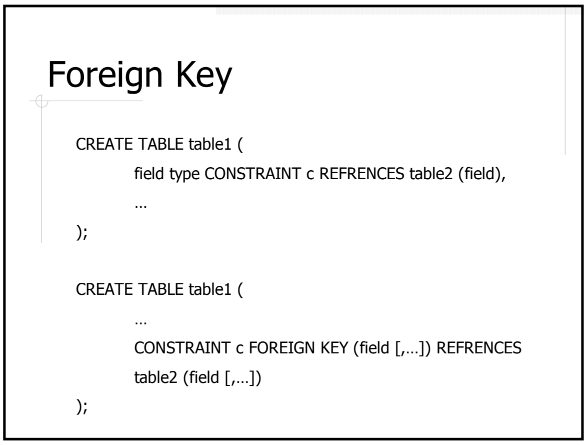
Not NULL

```
CREATE TABLE table (

        field type,
        …
        field type NOT NULL,
        …
        field type
);
```



Primary Key and Unique

```
CREATE TABLE table (
        field type CONSTRAINT c { PRIMARY KEY | UNIQUE },
        …
);

CREATE TABLE table (
        field type,
        …
        CONSTRAINT c { PRIMARY KEY | UNIQUE } (field [,…])
);
```



Foreign Key

```
CREATE TABLE table1 (
        field type CONSTRAINT c REFRENCES table2 (field),
        …
);

CREATE TABLE table1 (
        …
        CONSTRAINT c FOREIGN KEY (field [,…]) REFRENCES
        table2 (field [,…])
);
```