University of Warwick
Department of Computer Science

CS258

Database Systems

Cem Yilmaz
December 5, 2022

Contents

1 Introduction to Data

Data comes into two different kinds: structured and unstructured.

- Structured - e.g., tables with predefined columns
- Unstructured - e.g., web pages, text docs, images, videos...

Semi-structured data also exists and is found within XML and JSON. Traditionally, these are structured databases and were defined to be so.

Definition 1.1. Relation

Informally, a relation is a table of values having:

- A set of rows. The data elements in each row represent certain facts that correspond to a real-world entity or relationship. In the formal modal, rows are called tuples.
- Each column represents a characteristic / attribute of interest of that entity. Has a column header that gives an indication of the meaning of the data items in that column. In formal model, column header is called an attribute name.

2 Relational Model

2.1 Key

Each row must be uniquely identifiable in the table. The key of the row does this. Sometimes row-ids or sequential numbers are assigned as keys to identify the rows in a table. This is called an artificial key or surrogate key.

2.2 Table

A table is just an acceptable visual representation of the mathematical notion of relation. To formulate queries, we specify that table name(s), and attributes names of interest and special constraints (aka predicates) that need to be satisfied in order for a data item (=row) to be of interest.

2.3 Formal Definition

The schema of a relation is denoted by

$$R(A_1, A_2, \ldots, A_n)$$

where R is the name of the relation
The attributes / columns of the relations are denoted by $A_1, A_2, \ldots, A_n$. For example,

$$CUSTOMER(CUST-ID, CUST-NAME, ADDRESS, PHONENO)$$

Each Attribute/column has a domain or a set of valid values. For example, CUST-ID is a 7 digit number.
A tuple (aka row) of a relation is an ordered set of values enclosed in angled brackets $< \ldots >$. Each value is derived from an appropriate domain. For example,

$$< 632895, \text{"PETER T.", "2 Main St. Warwick", "(024)894} - 2000\text{"} >$$

A domain has a logical definition in the real world. A domain also has data format. For example, USA_phone_numbers may have a format: $(ddd)ddd - dddd$ where each d is a decimal digit. The attribute name designates thee role played by a domain in a relation. Used to interpret the meaning of the data elements corresponding to that attribute. Example, the domain Date may be used to define two attributes named "invoice-date" and "payment-date" with different meanings.

Definition 2.1. Relation State

The relation state R is the set that contains all the set of tuples in the relation. The relation state is a subset of the Cartesian product of the domains of its attributes. To put it all together,

$$\begin{aligned}
R(A_1, A_2, A \ldots, A_n) &\text{ is the schema of the relation} \\
R &\text{ is the name of the relation} \\
A_1, A_2, \ldots, A_n &\text{ are the attributes (columns) of the relation} \\
r(R) &\text{ is a specific state or instance of relation} \\
R &\text{ is an actual set of tuples (rows)} \\
r(R) = \{t_1, t_2, \ldots, t_n\} &\text{ where each } t_i \text{ is an } n\text{−tuple} \\
t_i =< v_1, v_2, \ldots, v_n > &\text{ where each } v_j \text{ comes from } dom(A_j) \\
r(R) \subset dom(A_1) \times dom(A_2) \times \ldots \times dom(A_n)
\end{aligned}$$

To create a table, we would write

```sql
CREATE TABLE Drinkers (
  name CHAR(31) Primary Key,
  Addr CHAR(50) DEFAULT '123 Sesame St',
  phone CHAR(16) NOT NULL
);
```

Listing 1: Creating a table

The constraints include the fact that there are key constraints, entity integrity constraints and referential integrity constraints. Also there exist domain constraints, values in an attribute in a tuple must come from the domain of that attribute. Values could also be NULL if allowed.

2.4 Key Constraints

Superkey of R : a subset of attributes of R, SK, such that:
In any valid state for $r(R)$: for all two distinct tuples t_1 and $t_2 \in r(R)$, where $t_1[SK] \neq t_2[SK]$, where SK is the superkey. For example, let us define superkey SK

$$SK = \{\text{Country, PhoneNumber}\}$$

then, no two entries country and phone number coincides.
A candidate key of R is a minimal super for any key K if:

- The removal of any attribute from K results in a set of attributes that is no longer a super key.

Definition 2.2. Superkey

Superkey is for all two tuples t_n and t_k in $r(R)$, there does not exist $t_n[SK] = t_k[SK]$.

Definition 2.3. Candidate Key

Otherwise known as the minimal key, it is by definition a super key that for which, if any attribute is removed from its definition of SK, it is no longer a super key.

2.5 Entity Integrity Constraints

The primary key PK cannot be NULL in any tuple of $r(R)$. This is because primary key values are used to identify the individual tuples. If PK has several attributes, null is not allowed as a value in any of these attributes. Also, any attribute of R even non-key may not be allowed to be NULL.

2.6 Referential Integrity Concerns cross-table relationships

A table students lists data about students, such as age, grades, etc. Students has an attribute STUDENTID, and suppose it is the primary key. Then, another table student-courses has also an attribute STUDENT ID, listing, for each module, which student are taking it, at which term etc. One could argue that all this info should be in one table, however, that is not a good idea and will be explained later. A logical DB Design spreads this information across tow tables, it is fundamental to R-DBMS design. Tables are "linked" according to references between them. Care must be exercised which such cross-table relationships are exercise.d.

2.7 Referential Integrity Constraints

A set of attributes FK from relation R_1 is a foreign key references relation R_2 : Attributes in the FK from R_1 have the same domain as the attributes of in the primary key PK of R_2, and a value of FK in a tuple t_1 of R_1 must either:

- refer to a value of the PK of some tuple t_2 in R_2,
- be NULL

Formally,

- t_1 in R_1 references t_2 in R_2 if $t_1[FK] = t_2[PK]$ or
- t_1 in R_1 makes no reference if $t_1[FK] = NULL$

Table 1: R_1

PK	FK
1	1
2	NULL
3	2

Table 2: R_2

PK	data
1	alpha
2	beta
3	gamma

Where in this case. $1 \rightarrow 1$, $2 \rightarrow 2$ from R_1 FK to R_2 PK. All of these constraints are expressed in create table in SQL. There also exist "semantic attribute integrity constraints" where, for example, things like bank balance > 0 are expressed.

2.8 Possible Violations for each operation

- Domain - one of the attribute values for new tuple is not in the attribute domain
- Key - the value of a key attribute in new tuple already exists
- Referential integrity - a foreign key value in new tuple references a primary key value that does not exist in referenced relation
- Entity integrity - if primary key value is null in new tuple.
- Delete - may violate only referential integrity, if primary key value of the tuple being deleted is referenced from other tuples in other relations. Some option must be specified during database design for each foreign key constraint on how to handle such deletions leading to relational integrity violations.
- UPDATE - obviously the domain and NOT NULL constraints may be violated on an attribute being modified. Furthermore, updating the primary key duplicates, updating a foreign key may violate referential integrity. Updating an ordinary attribute (not PK or FK) can violate only domain constraints.

2.9 Functional Dependencies

We need functional dependencies as they are a formal tool that allow us derivation of 'good' DB designs (relations and their schemata). Ultimately, a good design depends on the dependencies between attributes between to be in the same relation. Assume, X, Y, Z represent sets of attributes. A, B, C represent single attributes. Notation ABC implies $\{A, B, C\}$. $X \rightarrow Y$ is an assertion about a relation R and two sets of attributes from R.
If $X \rightarrow Y$ the values of the Y component of a tuple depend on values of the X component. i.e., the values of the X component of a tuple uniquely (functionally) determine those of the Y component.
So, $X \rightarrow Y$ specifies that whenever two tuples R agree on values of all the attributes of X, they must also agree on values of attribute of Y. Formally, $t_1[X] = t_2[X] \rightarrow t_1[Y] = t_2[Y]$.

2.10 Subsection

A db contains one or more schemas. Each schema contains one or more tables. To Create a schema, we type

CREATE SCHEMA company AUTHORIZATION 'Jsmith';

Creating a schema "company" owned by "Jsmith".