

SQL: Creating, updating, and modifying tables

Data Types in SQL (2.3)

- Character strings:
 - CHAR(n): fixed-length string of n characters.
 - VARCHAR(n): string of length of up to n characters.
- Bit strings:
 - BIT(n): bit string of length n.
 - BIT VARYING(n): bit string of length upto n.
- BOOLEAN: possible values are TRUE, FALSE, and UNKNOWN (read Chapter 6.1.6 and 6.1.7).
- Integers: INTEGER (INT), SHORTINT.
- Floats: FLOAT (or REAL), DOUBLE PRECISION.
- Fixed point numbers: DECIMAL(n, d): a number with n digits, with the decimal point d positions from the right.
- Dates and times: DATE and TIME (read Chapter 6.1.5).

Creating and Deleting Tables

- A table is a relation that is physically stored in a database.
- A table is **persistent**; it exists indefinitely unless deleted (dropped) or altered in some way.

Creating/Deleting Tables

- CREATE TABLE *name* (
 attr1 type1, attr2 type2, ...);

- DROP TABLE *name*;

Modifying Table Schemas

- ALTER TABLE name ADD attrib type;
- ALTER TABLE name DROP attrib;

Null and Default Values

- SQL allows NULL for unknown attribute values. (Read Chapter 6.1.6, especially for how SQL treats comparisons using NULL).
- NULL not allowed in certain cases.
- We can specify a default value for an attribute using the DEFAULT keyword.
 - CREATE TABLE *name* (*attrib type* DEFAULT *const*);
 - ALTER TABLE *name* ADD *attrib type* DEFAULT *const*;

Inserting Data into a Table (6.5)

- `INSERT INTO R(A1,A2, . . . An)`
`VALUES (v1, v2, . . . , vn);`
 - (A1, A2, . . . , An) can be a subset of R' s schema, or left out entirely.
 - Remaining attributes get NULL or DEFAULT values.
 - If attribute names left out, then you must provide values for all attributes and list values in standard order.

Inserting Data into a Table

- `SELECT INTO table(A1,...,An) Q;`
- Runs query Q and puts data into a relation.

Deleting Data from a Table

- `DELETE FROM R WHERE C;`
- Every tuple satisfying the condition C is deleted from R .

Updating Data in a Table

- `UPDATE table_name`
`SET column1=value1,column2=value2,...`
`WHERE some_column=some_value;`

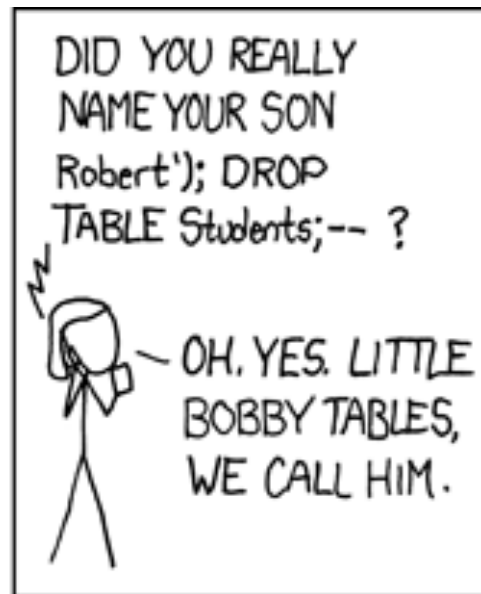
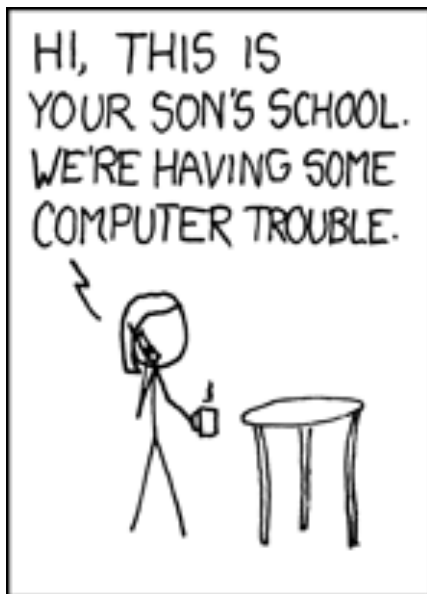
Loading/saving data in bulk

- Import/export data from/to text files in various formats.
 - Most common is comma separated values (CSV)
- Each RDBMS uses different syntax for this.

Constraints in Relational Algebra and SQL

Maintaining Integrity of Data

- You are creating a search engine for Rhodes' website, called Rhoogle.
- You have an SQL query:
 - "SELECT * FROM pages WHERE name=" + VAR + "";"



Maintaining Integrity of Data

- Data is **dirty**.
- How does an application ensure that a database modification does not corrupt the tables?
- Two approaches:
 - Application programs check that database modifications are consistent.
 - Use the features provided by SQL.

Integrity Checking in SQL

- PRIMARY KEY and UNIQUE constraints.
 - FOREIGN KEY constraints.
 - Constraints on attributes and tuples.
 - Triggers (schema-level constraints).
-
- How do we express these constraints?
 - How do we check these constraints?
 - What do we do when a constraint is violated?

Keys in SQL (2.3.6)

- A set of attributes S is a key for a relation R if every pair of tuples in R disagree on at least one attribute in S .
- Select one key to be the **PRIMARY KEY**; declare other keys using **UNIQUE**.

Primary Keys in SQL

- Creating a table with a primary key:
- `CREATE TABLE name (
 attrib1 type1 PRIMARY KEY,
 attrib2 type2, ...);`
or
`CREATE TABLE name (
 attrib1 type1, ...,
 PRIMARY KEY (attrib1, attrib2, ...));`

Effect of Declaring PRIMARY KEYS

- Two tuples in a relation cannot agree on all the attributes in the key. DBMS will reject any action that inserts or updates a tuple in violation of this rule.
- A tuple cannot have a NULL value in a key attribute.

Other Keys in SQL

- If a relation has other keys, declare them using the UNIQUE keyword.
- Use UNIQUE in exactly the same places as PRIMARY KEY.
- There are two differences between PRIMARY KEY and UNIQUE:
 - A table may have **only one PRIMARY KEY** but more than one set of attributes declared UNIQUE.
 - A tuple **may have NULL values in UNIQUE** attributes.

Enforcing Key Constraints

- Upon which actions should an RDBMS enforce a key constraint?
- Only tuple update and insertion.
- RDBMS searches the tuples in the table to find if any tuples exists that agrees with the new tuple on all attributes in the primary key.
- To speed this process, an RDBMS automatically creates an efficient search *index* on the primary key.
- Users can instruct the RDBMS to create an *index* on one or more attributes (more in Chapter 8.3, coming later).