Relational database languages

data manipulation language (DML)

data definition language (DDL)

SQL incorporates both

consistendy constraints

domain constraints

referential integrity

assertions

autorization

database system is a collection of databses

database is a collection of schema

schema a collection of inter-related tables

terminology is “flexible”

columns in different tables in a schema with same name should have the same domain

SQL: widely used non-procedural language

ids are synthetic keys – pretty much always unique

natural keys (like a name) may not be unique

inner joins are implied

(also have outer join and natural join)

natural join looks for all columns with the same name and joins on those

application programs generally access databases through:

- ~~language extensions to allow embedded sql~~

- application program interface (e.g., JDBC/ODBC) which allow SQL queries to be sent to a database

best practice is have a file in source code where all sql strings are stored and referenced from there

could also store sql in tables as procedures (best best practice)

create table is not the same as class declaration. It is more like

*List<Instructor> inst = new ArrayList<>();*

Database Design

- Unlike most modern software development processes, database design follows the “waterfall” methodology instead of the “agile” methodology.

- Schemas are “hardened” and resist CDO. (changes in schemas)

- Logical design – what data does the organization need to satisfy requirements? How should we organize the data into different entities

- Physical Design – not really relevant?

Normalization Theory

- more about this later

- noSQL is usually denormalized to make it easier

Entity Relationship Model

- like UML

- more about this later

- E-R relationship (between tables) != Relational Model relation (table)

see d2.

Normalization

scope of updates (how many things need to change if a value is updated)

many normal forms (1st 2nd 3rd 4th Boyscott)

Problem with example table is repetition of department info

Database Architecture

- how it works internally

Storage Management

- Storage manager is a program module that provides the interface between the low-level data store in the database and the application programs and queries submitted to the system.

- The storage manager is responsible to the following tasks:

- Interaction with the file manager

- Efficient storing, retrieving, and updating of database

Bigger companies (Oracle) access disk directly

- Issues

- indexing and hashing

b-Tree is used for sequential indexing

hashing is used strictly for random access

see d3.

Transaction Management

What if system fails?

Commit model vs rollback model

steps:

- mark beginning of transaction

- hit commit

- if commit doesn’t happen in a certain amount of time, it is rolled back

has an undo journal:

- marks entries as reversable. If something fails (no commit) will go back and undo entries marked

concurrent updates?

Pessimistic locking vs optimistic locking

- pessimistic locking will select rows and lock them so no one can touch while transaction is occurring

- optimistic locking adds a version number column. Only performs update if version number is still the same. Other wise it fails. (occurs somewhat in application code). Faster

Transaction-managerment component ensures database remains consistent (commit/rollback)

concurrency-controll manager controls the interatcion amon the concurrent transactions, to encure the consistency of the database (also deals with pessimistic locking).

View DSC 1.9 for Database system internals

- EDITS:

DDL puts stuff into data dictionary

compiler isn’t in databases

Database Application Architectures

Two-Tier vs Three-Tier

see DSC 1.6

internet applications are almost always 3-T

REST (Representational State Transfer) Post Get Put Delete kinda matches up with crud

History of Database Systems:

1960: DBMS emerges

1970: Relational DB

- Edgar Frank “Ted” Codd

IBM vs Codd

- IBM bet on IMS (inertia)

- Codd bet on relational DB

- Eventially 2 relational prototypes emerge

Ingres

- Build at UC Berkley

- Uses QUEL

- founcation for SYBase & MicrosoftSql

System R

- build at IBM

- leads to SEQUEL.. later SQL

- Evoleved to SQL/DS which evolved into db2

- project concludes that relational model is viable

Oracle

Larry Ellison watches IBM

Starts relational software

Entity Relationship first proposed in1976 by Peter Chen