Assertions, Views, and Programming

CS157A

Chris Pollett

Oct. 31, 2005.

Outline

- Assertions
- Views
- Database Programming

Assertions

- It is useful to be able to specify general constraints in SQL -- i.e., other than domain and key constraint.
- SQL allows you to create such constraints with CREATE ASSERTION:

```
CREATE ASSERTION SAL_CONSTRAINT
CHECK (NOT EXISTS
        (SELECT * FROM EMPLOYEE E, EMPLOYEE M,
           DEPARTMENT D WHERE E.SALARY > M.SALARY AND
           E.DNO=D.DNUMBER AND D.MGRSSN=M.SSN);
```

- If a tuple in the database causes such an assertion to evaluate to false, the assertion is **violated**.
- If no violating tuple exists then the assertion is said to be **satisfied**.

Oracle and Assertions

- Oracle does not support CREATE ASSERTION.
- However, the same effect can largely be achieved using triggers:

```
CREATE TRIGGER EMPLOYEE_TRIG
BEFORE INSERT OR UPDATE OF SALARY ON EMPLOYEE
-- Could also use AFTER rather than BEFORE
FOR EACH ROW -- this means for each row to be inserted
DECLARE
      MSAL INTEGER;
BEGIN
    SELECT M.SALARY INTO MSAL FROM EMPLOYEE M, DEPARTMENT D
    WHERE :new.DNO=D.DNUMBER AND D.MGRSSN=M.SSN
    IF (:new.SALARY > MSAL) THEN
      RAISE_APPLICATION_ERROR(-1234, 'Salary not allowed to be bigger than
    manager salary');
  END IF;
END;
```

- Notice :new and :old are special variables which refer to the tuple before or after the update.

Views

- Recall a **view** in SQL terminology is a single table that is derived from other tables.
- It is sometimes also called a **virtual table**.
- To create a view we use the syntax CREATE VIEW *name* AS *query*:

```
CREATE VIEW WORKS_ON1
    AS SELECT FNAME, LNAME, PNAME, HOURS
    FROM EMPLOYEE, PROJECT, WORKS_ON,
    WHERE SSN=ESSN AND PNO=PNUMBER;
```

- To delete a view we can use: DROP VIEW name;

View Implementation

- How are views handled by DBMS?
- Two possible approaches are:
  - **query modification** -- queries to the view are rewritten in terms of the base tables. Can lead to complex queries which are slow to evaluate
  - **view materialization** -- involves creating temporary view tables when the view is first queried and keeping this for future queries. If the view is not queried for a certain length of time it is deleted. For this to work need an efficient way to update a view when the base tables change.

View Update

- What should be done if someone wants to do an insert or an update on a view?
- If the view is on a single table, contains a primary key of that table, has all other attributes which are not null constraints, and does not involve aggregates, one can map the update on the view to the base table.
- Views on multiple tables or involving aggregates in general cannot be updated
- One can add the WITH CHECK OPTION to a CREATE VIEW declaration to indicate this view will be updated.

Database Programming

- We now begin discussing how to write canned transaction, i.e., **database applications**, which can be used by naïve users with little knowledge of how databases work.
- Such application are often written so as to be accessible via a web interface.

Approaches to Database Programming

- There are several different approaches to database application development:
  - *Embedding database commands in a general-purpose programming language*. A precompiler is then run on the source code to convert this code to code that makes function calls to the DBMS API library. Ex: proc or sqlj
  - *Using a library of database functions*. Here a library of functions is made available to the host programming language for database calls. This library might have functions calls for performing insert, updates, delete, queries, etc. Also known as using a database **Application Programming Interface** (API). Ex:Oracle OCI calls.
  - *Design a brand new language*. A database programming languages might be designed from scratch to handle the the database model and query language. Additional operations like looping, etc are added to make it Turing complete. Ex: PL/SQL

Impedance Mismatch

- One problem with the first two approached is impedance mismatch.
- **Impedance Mismatch** refers to the problems which occur because of differences between the database model and the programming language model.
- For example, databases deal with attribute, tuple, and tables.
- These have to be mapped to types in the host language.
- Not all host languages support the same types. So will need a different mapping for C, JAVA, etc.
- Also, the output of SQL queries are sets or multisets. So one need to set up a host language specific way to iterate over this kind of output.

Typical Sequence of Interaction in Database Programming

- Our database application will typically be written as a client program in a client server model.
- A common sequence of interactions is that:
  - The client program establishes a connection with the database server.
  - Once the connection is established, the program submits queries updates, or other database commands.
  - When the program is done, it should terminate or close the connection.

Embedded SQL, SQLJ, and JDBC Example

`See CS157b Fall 2003 HW2`