Databases

dr. Jörg Verstraete

Polish Academy of Sciences
Systems Research Institute

jorg.verstraete@ibspan.waw.pl
http://www.mini.pw.edu.pl/~verstraetej/
Today

• Lectures
  – SQL: DDL
    • simple create table statements
    • deletion of tables
    • modification of tables
  – SQL: DML
    • simple queries

• Labs
  – Exercises on SQL
SQL

• SQL: Structured Query Language
  – Querying data, updating data
  – Defining the datastructures
• Standard, with different revisions
  – SQL99 (=SQL3), SQL92 (=SQL2)
  – SQL86 (=SQL1), developed by ANSI, ISO
SQL standards:
- initially based on the common set of features implemented in IBM and Oracle
- only partly implemented by some DBMS vendors
- writing SQL code that is portable among different DBMS can be difficult
- different DBMS vendors offer extensions:
  - procedural statements like PL/SQL by Oracle
  - Transact-SQL by Microsoft

SQL3 defines:
- Object-oriented extensions,
- Core specification that should be implemented by every vendor of RDBMS,
- Optional packages addressing the needs of data warehouses, spatial data, time-based data, multimedia and other fields.

- SQL:2003: SQL for XML, generated columns, identity columns, ...
- SQL:2008: further extensions like INSTEAD OF triggers
SQL: DDL vs. DML

• DDL
  - Data Description (Definition) Language
    • Part of SQL that relates to the schema and datastructures, not to the content
      - Tables (definitions, ...)
      - Constraints
      - Indexes
      - Databases and tablespaces

• DML
  - Data Manipulation Language
    • Part of SQL that relates to entering, modifying, deleting and querying data
Create table / Alter table

• Create table: given name, including also
  – columns with datatypes
  – null/not null specification
  – key information (primary/foreign)
  – indexing information
  – …

• Alter table: given name, allows to
  – add/remove/modify columns
  – modify properties (null, key, indexing)
  – not always possible, depending on columns
Drop table / others

- Drop table
  - removes a database

- Additional
  - create database, drop database
  - create procedures, triggers, ...

- Comments
  - all actions bound to constraints
  - includes commands on DBMS specific implementation
Create table

• Syntax

CREATE TABLE
   [ database_name . [ schema_name ] . | schema_name . ] table_name
   ( { <column_definition> }[ , ...n ] )

<column_definition> ::= column_name <data_type>
...
   [ CONSTRAINT constraint_name ] DEFAULT constant_expression ]
...
   [ NULL | NOT NULL ]
...
   [ <column_constraint> [ ...n ] ]
...

<column_constraint> ::= [ CONSTRAINT constraint_name ]
{     { PRIMARY KEY | UNIQUE } [ CLUSTERED | NONCLUSTERED ]
...
| [ FOREIGN KEY ]
   REFERENCES [ schema_name . ] referenced_table_name [ ( ref_column ) ]
   [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }]
   [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
...
}
Alter table

• Syntax

ALTER TABLE [ database_name . [ schema_name ] . | schema_name . ] table_name
{
  ALTER COLUMN column_name
  |
  <column_definition>
 |
  DROP
 |
 | COLUMN
 |
 |
}
DDL

- Strongly depends on DBMS
  - different options for different DBMS

- Subject to constraints
  - referential integrity: cannot remove a column that participates in a relation
  - not all operations possible
    - e.g. changing the type of an attribute
    - e.g. changing primary key