MTAT.03.105
Introduction to Databases

Lecture #2
Introduction to SAP SQL Anywhere 17

Ljubov Jaanuska
(ljubov.jaanuska@ut.ee)
Lecture 1. Summary

Data is a collection of unprocessed items that may consist of text, numbers, images and video.

Information refers to the data that have been processed in such a way that the knowledge of the person who uses the data is increased.

DBMS is a software system that enables users to define, create, maintain, and control access to the database.

A database is a shared collection of logically related data, and a description of this data, designed to meet the information needs of an organisation.
Lecture 2. What will you learn

- Types of primary SQL statements
- Create a database
- Create a table
- INSERT, SELECT, UPDATE, DELETE rows
- SQL Anywhere documentation
- DB development process
- PCA
# Types of SQL statements

SQL language is divided into five types of primary language statements: DML, DDL, DCL, TCL, and DRL.

<table>
<thead>
<tr>
<th>Data Definition Language</th>
<th>Data Manipulation Language</th>
<th>Data Control Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. create</td>
<td>1. insert</td>
<td>1. Grant</td>
</tr>
<tr>
<td>2. alter</td>
<td>2. update</td>
<td>2. Revoke</td>
</tr>
<tr>
<td>3. drop</td>
<td>3. delete</td>
<td></td>
</tr>
<tr>
<td>4. truncate</td>
<td>4. merge</td>
<td></td>
</tr>
<tr>
<td>5. rename</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transaction Control Language</th>
<th>Data Retrieval Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Commit</td>
<td>1. Select</td>
</tr>
<tr>
<td>2. Rollback</td>
<td></td>
</tr>
<tr>
<td>3. Savepoint</td>
<td></td>
</tr>
</tbody>
</table>
Using these statements, we can define the structure of a database by creating and altering database objects, and we can manipulate data in a table through updates or deletions. We also can control which user can read/write data or manage transactions to create a single unit of work.

### What are DDL, DML, DCL, TCL and DRL commands?

<table>
<thead>
<tr>
<th>Data Definition Language</th>
<th>Data Manipulation Language</th>
<th>Data Control Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.create</td>
<td>1.insert</td>
<td>1.Grant</td>
</tr>
<tr>
<td>2.alter</td>
<td>2.update</td>
<td>2.Revoke</td>
</tr>
<tr>
<td>3.drop</td>
<td>3.delete</td>
<td></td>
</tr>
<tr>
<td>4.truncate</td>
<td>4.merge</td>
<td></td>
</tr>
<tr>
<td>5.rename</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transaction Control Language</th>
<th>Data Retrieval Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.Commit</td>
<td>1.Select</td>
</tr>
<tr>
<td>2.Rollback</td>
<td></td>
</tr>
<tr>
<td>3.Savepoint</td>
<td></td>
</tr>
</tbody>
</table>
Basic rules for SQL statements

- No commands, only statements
- SQL is not case sensitive language
- Use semicolon (;) at the end of each statement
- For readability write SQL keyword on separate line
- It is good practice type SQL keywords in capital letters
Create a database

- Start **SAP Central** and create a new database:
  - *Tools* -> *SQL Anywhere 17* -> *Create Database*

- Use default settings apart from:
  - Specify the database name as *PCA* and save it into a separate folder
  - Specify username as *DBA* and password as *sql* (only for educational reasons and let the instructor check the assignments)
  - For alphanumeric sorting choose the *UTF-8* collation which is not case sensitive (twice)
  - On the *Security Model* page choose compatibility to connect to earlier SQL Anywhere versions

- Pay attention to *.log file* (leave it unchanged)
Connect to the database

- Start **client** (*e.g.* Central)
- Log in as DBA (psw: sql)
- Choose action: **Start and connect a database on this computer**
- Browse for PCA.db
- On connection to database, **server** starts automatically
- If any **client** is connected to database (*e.g.* Central) and **server** is running, another **client** can be connected using the username and password, the action should be as **Connect to running database on this computer**
SQL Anywhere files

• To be submitted:
  • 1 main file (.db)
  • 1 transaction log file (.log)
    It is not a text document!
  • To connect to a database, both files are needed (do not change .log file, otherwise the database will be corrupted)

• Optional:
  • All run statements can be stored in one file (.sql)
Names

• Server names must be unique within one network
• Database names must be unique within one server
Interactive SQL (iSQL)

- Interactive SQL – our main tool to learn SQL (it is a client with two windows)
- Run from Central
  - Tools -> SQL Anywhere 17 -> Open Interactive SQL
First table

Run iSQL and type table definition (DDL)

```
CREATE TABLE Person(
    name varchar(30) not null,
    surname varchar(100) not null,
    sex char(1) not null check (sex in ('m', 'f')),
    birthday date not null,
    entered datetime not null default current timestamp,
    personal_id varchar(11),
    CONSTRAINT pk_person PRIMARY KEY (personal_id)
);
```
Insert data

```sql
INSERT INTO Person
(name, surname, sex, birthday, personal_id)
VALUES
('Juku', 'Mets', 'M', '1980-02-04', '38002042715');
```

Add two people more!

```
SELECT * FROM Person;
```

```
SELECT name, surname FROM Person;
```
Select row from table

SELECT * FROM Person WHERE person_id= '...';
SELECT * FROM Person WHERE sex= 'm';
SELECT * FROM Person WHERE person_id= '...' OR person_id= '...';
SELECT * FROM Person WHERE person_id='...' AND person_id= '...';
Modify row in table

UPDATE Person SET name= 'Jüri' WHERE name= 'Juku';

INSERT INTO person(name, surname, sex, birthday, person_id)
VALUES ('Mati', 'Karu', 'M', '1985-02-04', '38002042715');

(an error message because person_id must be unique)
Delete row from table

```sql
DELETE person WHERE name= 'Juku';

DELETE person WHERE name= 'Jüri';
```

Important!!!

DELETE person;

(the statement will delete all data from table `person`).
For such purpose use TRUNCATE which operated faster
Sum up

CREATE TABLE
INSERT INTO
UPDATE
DELETE
DROP TABLE
SELECT
Delimiters in documentation

SQL Anywhere documentation: http://dcx.sap.com/index.html

Delimiters:
{} compulsory part of the statement
[] non-compulsary part of the statment
() expression
| alternatives (only one can be choosen)
... so on, so forth
:= definitions
Capital letters - SQL reserved words
Small letters – variables (names, literals)
CREATE LOCAL TEMPORARY TABLE
[ IF NOT EXISTS ]
[owner.]table-name
( { column-definition [ column-constraint ... ] | table-constraint | pctfree }, ... )
[ ON COMMIT { DELETE | PRESERVE } ROWS | NOT TRANSACTIONAL ]
Brain teaser

TRUNCATE TABLE [owner.]table-name | MATERIALIZED VIEW [owner.]materialized-view-name
DB development process

- Problems statement
- Analysis
- Schema development
- Views
- Application
- DB administration and development
Problem statement

- A database for Professional Chess Association (PCA)
- Data to track:
  - Personal data about players
  - Club data
  - Game data
  - Tournament data
- DB users are PCA members and fans
Problem statement

• Using the DB, the client wants to get answers to the following questions:
  • Who is the best player in Estonia?
  • Which club has the most players?
  • When and where did the tournament “Estonian Championship’13” take place?
  • Which color brings luck: black or white chess pieces?
Analysis

- **Entities**:
  - Player
  - Club
  - Game
  - Tournament

- **Important issues**
  - Names (alphabet, double names)
  - Id vs personal_id
  - Results
  - Venue