1 лекция devj130 (SQL)

/\*Сначала указать javadb в glassfish, а потом где лежит база.

DDL (Data definition language) – язык для модификации таблиц в базах данных.

Однострочный комментарий --

CREATE TABLE tablename (field-name type constraints, …);

Data types in SQL:

• INTEGER (int) – целочисленный тип

• DOUBLE – вещественный тип

• CHAR(length) – символьный тип

• VARCHAR(length) – символьный тип(строка) переменной длины

• BOOL(EAN) – логический тип TRUE, FALSE, NULL(не всеми версиями СУБД поддерживается)

• NULL – встроенная константа, которая означает неопределенное значение(это не 0)

Возможные ограничения:

• PRIMARY KEY – уникальный первичный ключ для таблицы

• FOREIGN KEY – внешний ключ(в таблице можем организовать ссылку на первичный ключ другой таблицы)

• NOT NULL – обязательность определенного значения

• DEFAULT значение – определение значения по умолчанию

Example creating table:

\*/

CREATE TABLE a (id INTEGER PRIMARY KEY, title VARCHAR(16) NOT NULL);

CREATE TABLE b (id INTEGER PRIMARY KEY, id\_a INTEGER, comment VARCHAR(255),

CONSTRAINT fk\_a FOREIGN KEY(id\_a) REFERENCES a(id));

/\*

Модификация таблицы:

ALTER TABLE name

Commands:

• ADD | MODIFY | DELETE COLUMN name values;

Example modify table:

\*/

ALTER TABLE a ADD COLUMN notes VARCHAR(255);

/\*

Удаление таблицы:

DROP TABLE name;

\*/

DROP TABLE a;

DROP TABLE b;

/\*

DML(Data Manipulation Language) – язык для работы с значениями таблицы.

Commands:

• INSERT INTO table (field-list) VALUES (value-list); -добавить значения

INSERT INTO a (id, title) VALUES (1, 'first record');

INSERT INTO a VALUES (2, 'second record');

• UPDATE table SET column-name=value, … WHERE condition; -заменить значение

UPDATE a SET title = ‘\*second record\*’ WHERE id = 2;

• DELETE FROM table WHERE condition; - удаление значения

DELETE FROM a WHERE id = 3;

• SELECT ALL|DISTINCT field-list | \* function FROM table

WHERE condition

HAVING condition

GROUP BY field-list – группировка значений

ORDER BY field-list ASC | DESC; вывод по возростанию или убыванию

DISTINCT – вывод только уникальных записей

Conditions:

=, >, <, <=, >=, NOT, LIKE

Field-name LIKE pattern: % \_

\*/

INSERT INTO a (id, title) VALUES (1, 'first record');

INSERT INTO a VALUES (2, 'second record');

INSERT INTO a VALUES (3, 'third record');

INSERT INTO a VALUES (4, ' record');

INSERT INTO b VALUES (1,1, 'comment to first record');

INSERT INTO b VALUES (2,1, 'second comment to first record');

INSERT INTO b VALUES (3,2, 'comment to second record');

UPDATE a SET title = '\*second record\*' WHERE id = 2;

DELETE FROM a WHERE id = 3;

/\*

Functions:

COUNT –подсчет полей

SUM – подсчет числовых значений

MIN | MAX – минимальное и максимальное

Вывод из двух таблиц:

SELECT … FROM left-table INNER | LEFT | RIGHT | CROSS JOIN right-table ON (condition)

\*/

SELECT \* FROM a;

SELECT \* FROM a WHERE id = 1;

SELECT \* FROM a WHERE title LIKE '\*s%';

SELECT title FROM a GROUP BY title;

SELECT COUNT(\*) AS Num, title FROM a GROUP BY title;

SELECT COUNT(\*) AS Num, title FROM a GROUP BY title HAVING COUNT(\*) > 1;

SELECT a.ID, a.TITLE, b.COMMENT FROM a INNER JOIN b ON (a.ID = b.ID\_A);

SELECT a.ID, a.TITLE, b.COMMENT FROM a LEFT JOIN b ON (a.ID = b.ID\_A);

SELECT a.ID, a.TITLE, b.COMMENT FROM b RIGHT JOIN a ON (a.ID = b.ID\_A);

Почитать как включить автозаполнение и подсказки в netbeans