

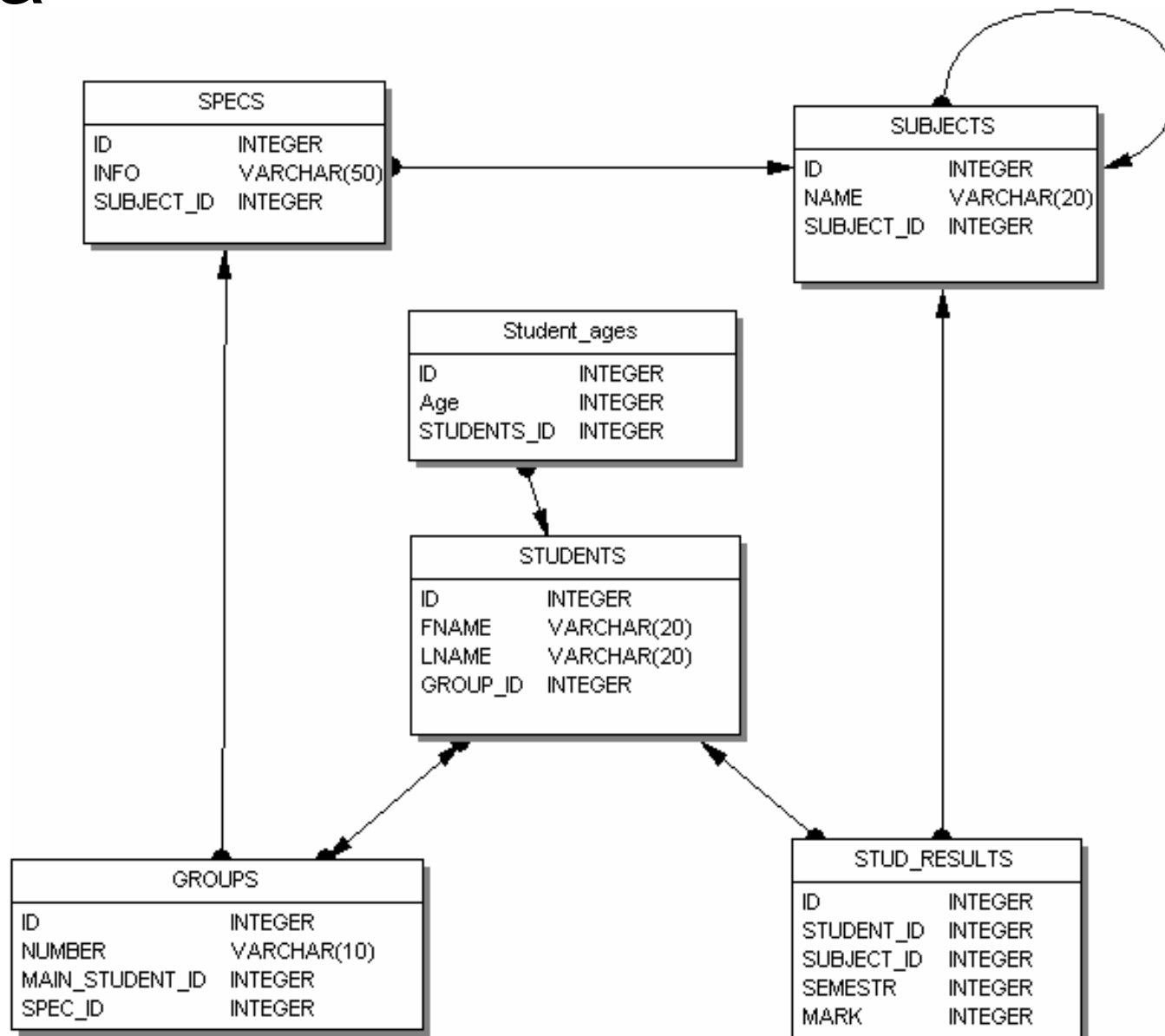
Лекция #4

Базы данных

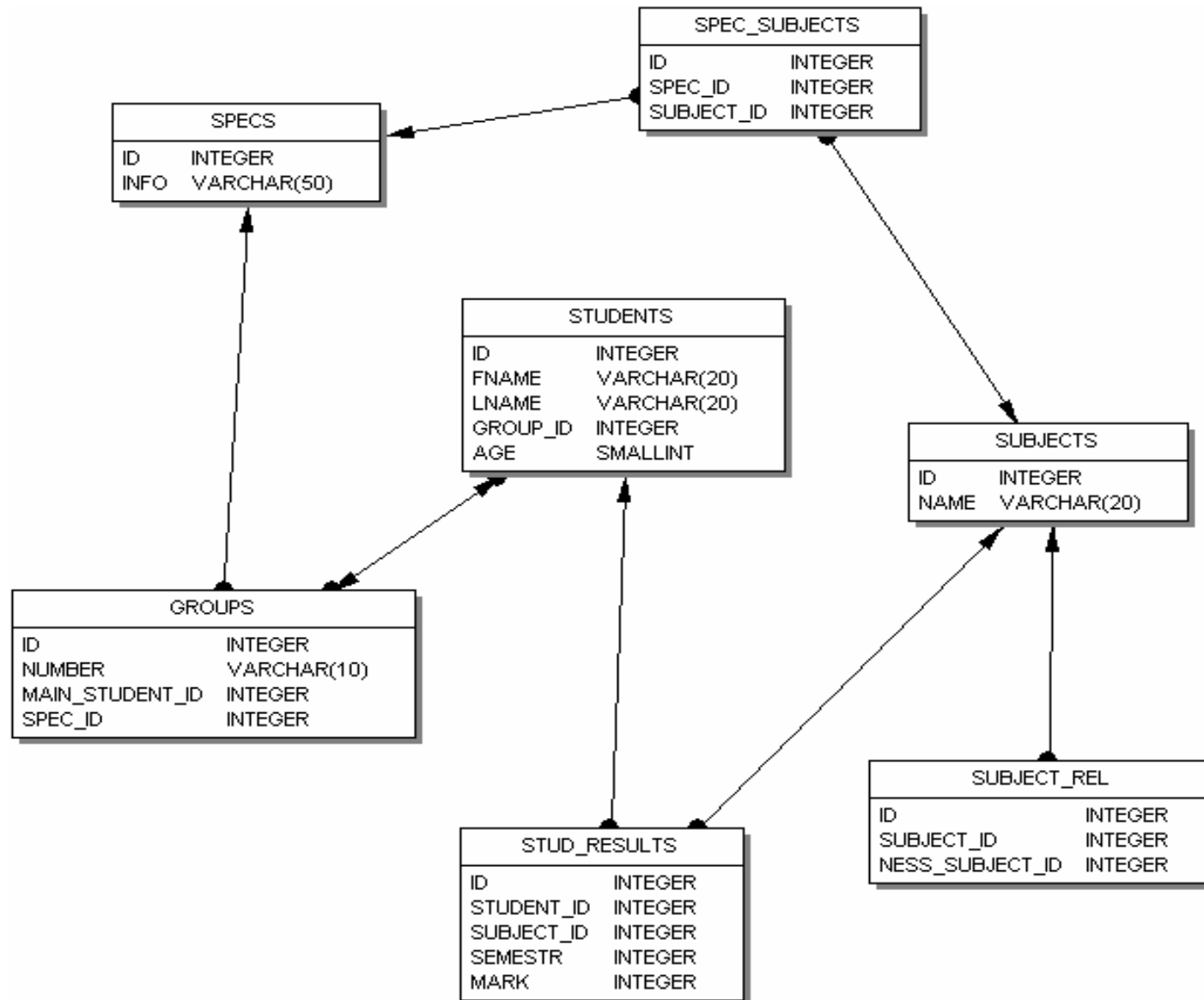
Михаил Моисеев

Structured Query Language

Задача



Решение



Определения

В SQL определены два подмножества языка:

- **SQL-DDL** (Data Definition Language) - язык определения структур и ограничений целостности баз данных.

Сюда относятся команды создания и удаления баз данных; создания, изменения и удаления таблиц; управления пользователями и т.д.

- **SQL-DML** (Data Manipulation Language) - язык манипулирования данными: добавление, изменение, удаление и извлечение данных, управления транзакциями

История и стандарты

- System R IBM язык SEQUEL 1970
- SQL/DS IBM 1981
- Первый стандарт SQL – ANSI и ISO 1986
- SQL dialect 1 – ANSI 1987
- SQL-92 – ANSI 1992
- SQL dialect 3 – ANSI 1999
- SQL-2003

Типы данных

Целые числа:

- INTEGER - обычное целое 4 байта;
- SMALLINT - короткое целое 2 байта.

Строки:

- CHAR(n) - символьная строка длины n;
- VARCHAR(n) - символьная строка переменной длины, но не более n;

Вещественные типы данных:

- FLOAT - число с плавающей точкой 4 байта.
- DECIMAL(p,n) - p - число знаков до запятой, n - после запятой;
- DOUBLE PRECISION - число с плавающей точкой 8 байт.

Другие типы данных:

- BLOB – двоичные типы данных большого размера;
- DATE – дата с точностью до дня, единица измерения день;
- TIME - время с точностью до мс, единица измерения секунда;
- TIMESTAMP – дата и время вместе.

SQL-DDL

Основные операторы управления метаданными:

- CREATE - создать;
- ALTER - изменить;
- DROP - удалить.

Выполняются над сущностями:

- DOMAIN
- TABLE
- VIEW
- PROCEDURE
- TRIGGER
- GENERATOR
- INDEX
- ...

Управление БД

```
CREATE DATABASE <путь к БД> USER <ЛОГИН>  
PASSWORD <пароль> [...];
```

```
CREATE DATABASE '10.1.12.4:/var/db/stud.fdb' USER 'SYSDBA'  
PASSWORD 'masterkey'  
PAGE_SIZE 16384 DEFAULT CHARACTER SET CYRL;
```

```
CONNECT DATABASE <путь к БД> USER <ЛОГИН >  
PASSWORD <пароль> [...];
```

```
DROP DATABASE;
```


Управление таблицами

CREATE TABLE <имя_таблицы> (< опр_поля >,)

<опр_поля> = <имя_поля>

{<тип_поля> | **COMPUTED BY** (<выражение>) | <домен> }

[**DEFAULT** {<литерал> | **NULL** | **USER** | ...}]

[**NULL** | **NOT NULL**]

[<огр_поля>]

```
CREATE TABLE students (id integer NOT NULL,  
                        fname varchar(30),  
                        lname varchar(30),  
                        yob integer DEFAULT 1985,  
                        age COMPUTED BY (2008-yob))
```

Управление таблицами #2

```
ALTER TABLE <имя_таблицы>  
{ADD <опр_поля>| DROP<имя_поля>},...
```

```
ALTER TABLE students  
  ADD hobby varchar(20) NOT NULL,  
  DROP group_id;
```

```
DROP TABLE <имя_таблицы>
```

```
DROP TABLE students;
```

Ограничения

Ограничения на значения полей таблицы, 4 типа ограничений.

```
<огр_поля> = [CONSTRAINT <имя_огр>]  
{ PRIMARY KEY |  
  UNIQUE |  
  FOREIGN KEY (<имя_поля>) REFERENCES  
  <имя_глав_таблицы>(<имя_поля>) |  
  CHECK <condition> }
```

Ограничения #2

Ограничения первичного ключа - **PRIMARY KEY**

```
CREATE TABLE students(id integer NOT NULL PRIMARY KEY,  
                        group_id integer NOT NULL ,  
                        fname varchar(30) NOT NULL,  
                        lname varchar(30),  
                        age integer);
```

```
ALTER TABLE students ADD CONSTRAINT pk1 PRIMARY KEY (id);
```

Ограничения #3

Ограничения уникальности - **UNIQUE**

```
CREATE TABLE students(id integer NOT NULL UNIQUE,  
                        group_id integer NOT NULL ,  
                        fname varchar(30) NOT NULL,  
                        lname varchar(30),  
                        age integer);
```

```
ALTER TABLE students ADD CONSTRAINT unique1 UNIQUE (id);
```

Ограничения #4

Ограничения на диапазон значений поля - **CHECK**

```
CREATE TABLE students(id integer NOT NULL,  
                        group_id integer NOT NULL ,  
                        fname varchar(30) NOT NULL,  
                        lname varchar(30),  
                        age integer CHECK (age > 18));
```

```
ALTER TABLE students ADD CONSTRAINT check1 CHECK (age > 18);
```

Ограничения #5

Ограничения на диапазон значений поля - **FOREIGN KEY**

```
CREATE TABLE students(id integer NOT NULL,  
                       group_id integer NOT NULL ,  
                       fname varchar(30) NOT NULL,  
                       lname varchar(30),  
                       age integer,  
CONSTRAINT fk_group FOREIGN KEY(group_id) REFERENCES groups (id) );
```

```
ALTER TABLE students ADD CONSTRAINT fk_group  
FOREIGN KEY(group_id) REFERENCES groups (id);
```

Вопросы

- Какие операторы используются для управления метаданными ?
- Какие типы ограничений на поля таблицы существуют ?
- Можно ли изменять структуру уже созданной таблицы ?