Операторы определения данных

Язык определения данных DDL

используется для создания и изменения структуры базы данных и ее составных частей – таблиц, индексов, представлений (виртуальных таблиц), а также триггеров и сохраненных процедур

Основные операторы ЯОД

CREATE (создать) **ALTER (модифицировать)** **DROP (удалить)**	✖	**DATABASE (базу данных)** **TABLE (таблицу)** **VIEW (виртуальную таблицу)** **INDEX (индекс)** **TRIGGER (триггер)** **PROCEDURE (сохраненную процедуру)**

Основные типы данных

1. **Символьные** – строки фиксированной и переменной длины
2. **Строки потенциально неограниченного размера**
3. **Целые числа**
4. **Вещественные числа**
5. **Булевы значения**
6. **Денежные значения**
7. **Время:**
8. **Двоичные данные**

Манипулирование таблицами

CREATE TABLE – создать таблицу

ALTER TABLE – модифицировать таблицу

DROP TABLE – удалить таблицу

Таблица –

основной объект для хранения информации в реляционной БД. Она состоит из содержащих данные строк и столбцов, занимает в БД физическое пространство и м.б. ***постоянной*** или ***временной***.

Основные ???

- Как будет называться таблица?
- Как будут называться столбцы (поля) таблицы?
- Какие типы данных будут закреплены за каждым столбцом?
- Какой размер памяти должен быть выделен для хранения каждого столбца?
- Какие столбцы таблицы требуют обязательного ввода?
- Из каких столбцов будет состоять первичный ключ?

Создание таблицы

CREATE TABLE имя_таблицы

(имя_столбца тип_данных

[DEFAULT выражение **] [** ограничения_столбца **],**

.....

[ограничения_таблицы **]**

)

[DEFAULT выражение **] –** значение по умолчанию

Ограничения столбца

NOT NULL – обязательность значений для столбца

UNIQUE – уникальность значений в столбце

PRIMARY KEY – первичный ключ

FOREIGN KEY – внешний ключ

CHECK(условие проверки на допустимость) – дополнительное ограничение на вид значений столбца

```
CREATE TABLE student
( fio      char(16)     not null      PRIMARY KEY,
  gr       char(2)      not null      CHECK( gr in (‘4А',‘4Б') ),
  spec     char(10)
)
```

```
CREATE TABLE student
( fio        char(16)    not null,
  gr         char(2)     not null,
  spec       char(10),
  PRIMARY KEY(fio),
  CHECK( gr in (‘4А',‘4Б') ),
)
```

```
CREATE TABLE uspevaemost
( fio      char(16)     not null    FOREIGN KEY references student(fio),
  predm    char(20)     not null,
  ocenka smallint
)
```

Для анализа ошибок целесообразно именовать ограничения, используя CONSTRAINT.

Общепринято сопровождать имя сокращенными обозначениями ограничений:

для первичного ключа –	PK
для внешнего ключа –	FK
для проверочного ограничения –	CK
для ограничения уникальности –	U
для значения по умолчанию –	DF

```
CREATE TABLE student
( fio          char(16)      not null,
  gr           char(2)   not null,
  spec      char(10),

  CONSTRAINT PR_student PRIMARY KEY(fio),
  CONSTRAINT CH_gr CHECK( gr in (‘4А',4Б') )
)
```

Создание таблицы на основе существующей

CREATE TABLE имя_таблицы
[(имя_столбца ограничения, …)]
AS подзапрос

- количество заданных столбцов должно совпадать с количеством столбцов в подзапросе,
- для столбцов можно указать только имя и ограничения
- в новую таблицу копируется только ограничение **NOT NULL**

Создание таблицы на основе существующей

SELECT имена_столбцов
INTO имя_таблицы
FROM имя_таблицы_источника
[WHERE условие **]**

```
SELECT  *
INTO СТУДЕНТ1
FROM СТУДЕНТ
```

копия структуры и данных в новую таблицу

```
SELECT  *
INTO СТУДЕНТ1
FROM СТУДЕНТ
WHERE Группа='0'
```

копия только структуры в новую таблицу (заведомо ложное условие)

```
SELECT  *
INTO СТУДЕНТ_5А
FROM СТУДЕНТ
WHERE Группа='5А'
```

создание таблицы с данными о студентах группы 5А

Изменение таблиц и ограничений

Обобщенный формат:

ALTER TABLE имя_таблицы
перечень фраз-изменений

Варианты изменений:

Добавить новый столбец:

ADD [COLUMN] имя_столбца тип_данных ограничения

Удалить столбец:

DROP [COLUMN] имя_столбца **[RESTRICT | CASCADE]**

RESTRICT – удаление отменяется, если существуют ссылки на удаляемый компонент (по умолчанию)

CASCADE – удаление осуществляется каскадным образом с удалением ссылок на удаляемый компонент

Варианты изменений:

Добавить новые ограничение:

ADD [**CONSTRAINT** [имя_ограничения]] ограничение

Удалить ограничение:

DROP CONSTRAINT имя_ограничения [**RESTRICT | CASCADE**]

Варианты изменений:

Задать значение по умолчанию:

ALTER [COLUMN] SET DEFAULT значение

Отменить значение по умолчанию:

ALTER [COLUMN] DROP DEFAULT

Удаление таблицы

DROP TABLE имя_таблицы [**RESTRICT | CASCADE**]

Системные типы данных MS SQL server

Числовые:

целое число

bit	1 бит
tinyint	1
smallint	2
int	4
bigint	8

точное вещественное число

decimal [(p[,s])] или numeric [(p[,s])]	5-17

Числовые:

приближенное вещественное число

real	4
float [(n)]	8

денежный

smallmoney	4
money	8

Дата, время:

smalldatetime	1 января 1900 – 6 июня 2079
datetime	1 января 1753 – 31 декабря 9999

Символьные:

не поддерживающие UNICODE

char [(n)]	0-8000
varchar [(n)]	0-8000
varchar(max)	0-2Гб или 16-разрядный указатель
text	0-2Гб или 16-разрядный указатель

поддерживающие UNICODE

nchar [(n)]	0-8000
nvarchar [(n)]	0-8000
nvarchar(max)	0-2Гб или 16-разрядный указатель
ntext	0-2Гб или 16-разрядный указатель

Системные типы данных SQL server

1. Символьные: CHAR(size), VARCHAR(maxsize) – строки фиксированной и переменной длины

2. Строки потенциально неограниченного размера:
TEXT

3. Целые числа:
BIGINT – 8-и байтовые,
INT – 4-х байтовые,
SMALLINT – 2-х байтовые
TINYINT – 1 байтовое положительное целое от 0 до 255

4. Вещественные числа:
FLOAT
REAL
DECIMAL(len,dec)

Основные типы данных

5. Булево значение: BIT – значение 0 или 1

6. Денежный: MONEY
SMALLMONEY

7. Время:
DATETIME – дата высокой точности
SMALLDATETIME – дата низкой точности

8. Двоичные данные:
BINARY(size), VARBINARY(maxsize) – фиксированной и переменной длины
IMAGE – файлы