

Схема базы данных

Подмножество DDL языка SQL

Метаданные

- ◆ Все объекты, определенные в базе данных называются *метаданными* и *определяют её схему*
- ◆ СУБД сохраняют метаданные в таблицах, которые создаются непосредственно в самой базе данных в разделе *системных таблиц* (принцип единообразия)
- ◆ Для управления схемой базы данных используется язык определения данных (DDL) – подмножество языка SQL

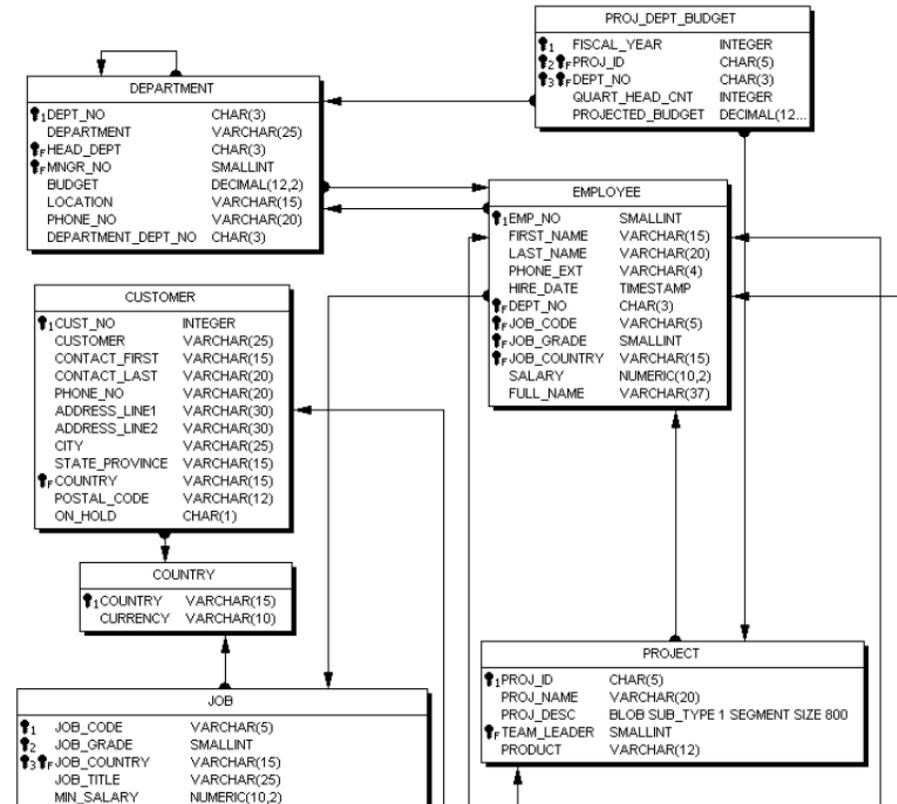
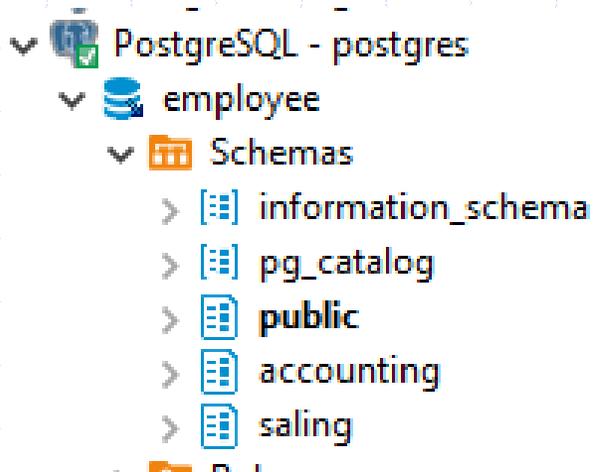
Метаданные

- ◆ Создание объекта схемы CREATE (INSERT в системные таблицы)
- ◆ Изменение объекта схемы ALTER (UPDATE и INSERT)
- ◆ Удаление объекта схемы DROP (DELETE и UPDATE)

БД и схема БД

- ◆ В чем разница?
 - Схема – структура БД

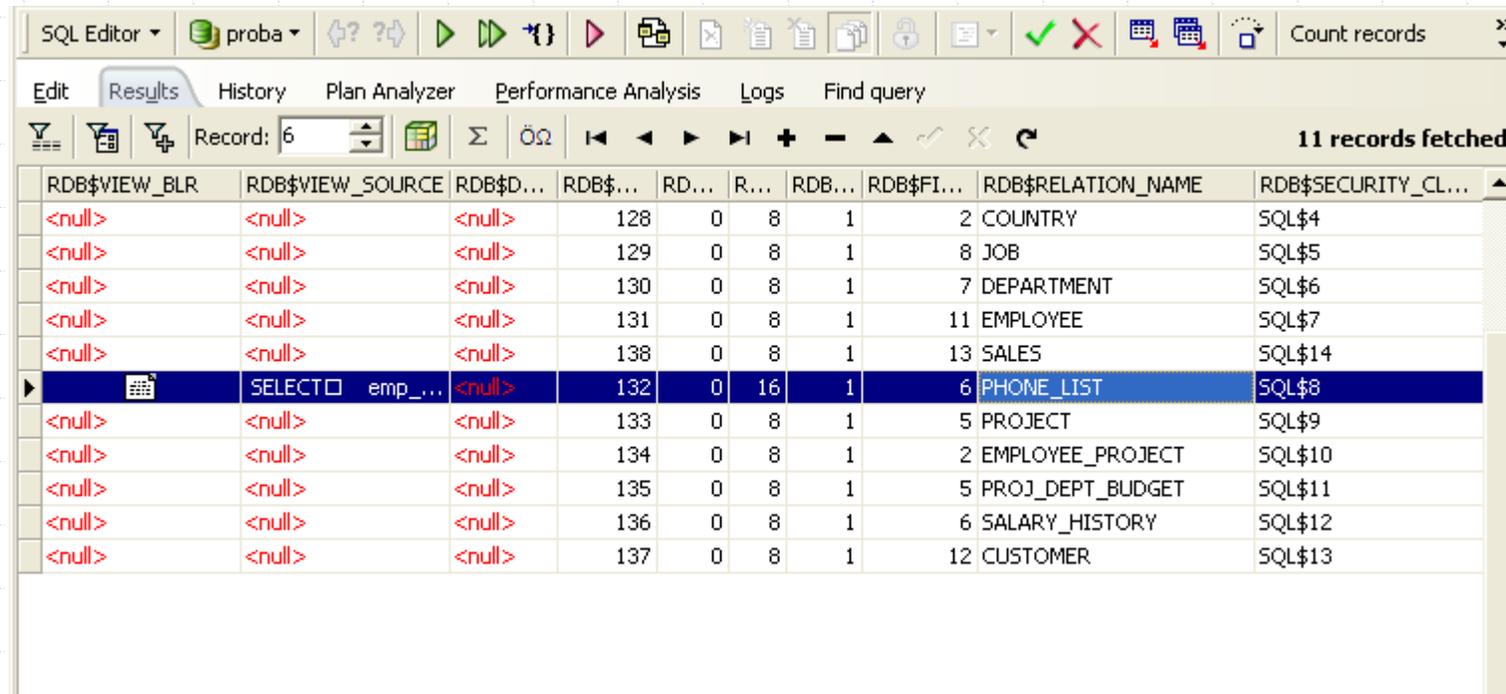
- Схема- объект БД



Системные таблицы

```
select *
```

```
from rdb$relations  
where rdb$flags=1
```



RDB\$VIEW_BLR	RDB\$VIEW_SOURCE	RDB\$D...	RDB\$...	RD...	R...	RDB...	RDB\$FI...	RDB\$RELATION_NAME	RDB\$SECURITY_CL...
<null>	<null>	<null>	128	0	8	1	2	COUNTRY	SQL\$4
<null>	<null>	<null>	129	0	8	1	8	JOB	SQL\$5
<null>	<null>	<null>	130	0	8	1	7	DEPARTMENT	SQL\$6
<null>	<null>	<null>	131	0	8	1	11	EMPLOYEE	SQL\$7
<null>	<null>	<null>	138	0	8	1	13	SALES	SQL\$14
	SELECT emp_...	<null>	132	0	16	1	6	PHONE_LIST	SQL\$8
<null>	<null>	<null>	133	0	8	1	5	PROJECT	SQL\$9
<null>	<null>	<null>	134	0	8	1	2	EMPLOYEE_PROJECT	SQL\$10
<null>	<null>	<null>	135	0	8	1	5	PROJ_DEPT_BUDGET	SQL\$11
<null>	<null>	<null>	136	0	8	1	6	SALARY_HISTORY	SQL\$12
<null>	<null>	<null>	137	0	8	1	12	CUSTOMER	SQL\$13

Создание БД (Firebird)

CREATE { DATABASE | SCHEMA }

'<спецификация файла>'

[USER '<имя пользователя> ' [PASSWORD '<пароль> ']]

[PAGE_SIZE [=] < целое >]

[LENGTH [=] < целое > [PAGE[S]]]

[DEFAULT CHARACTER SET < набор символов >]

[<вторичный файл>];

[<вторичный файл>] = FILE '<спецификация файла>'

[LENGTH [=] < целое > [PAGE[S]] | STARTING [AT [PAGE]] < целое >] [
вторичный файл >]

- Администраторы
- Пользователи с привилегией CREATE DATABASE

На сервере (IBExpert)

Create Database

Server: Remote | Server name: | Protocol: TCP/IP

Database:

Client Library File: C:\Program Files\Firebird\Firebird_2_1\bin\fbclient.dll

Username: | Password: | SQL Dialect: Dialect 3

Page Size: 16384 | Charset: NONE | Collation (FB 2.5):

Register Database After Creating

OK Cancel Help

Локально (Windows)

```
CREATE DATABASE 'D:\test.fdb'  
USER sysdba PASSWORD 'masterkey'  
DEFAULT CHARACTER SET WIN1251;
```

Create Database

Server / Protocol
Local, XNet (FB 3)

Database
D:\work\BD\samples.fdb

Connection string
xnet://D:\work\BD\samples.fdb

Client Library File
D:\work\BD\fbclient.dll

Username SYSDBA SQL Dialect Dialect 3

Password ***** Role

Page Size 16384

Charset WIN1251 Collation (FB 2.5)

Register Database After Creating

OK Cancel Help

Изменение схемы БД

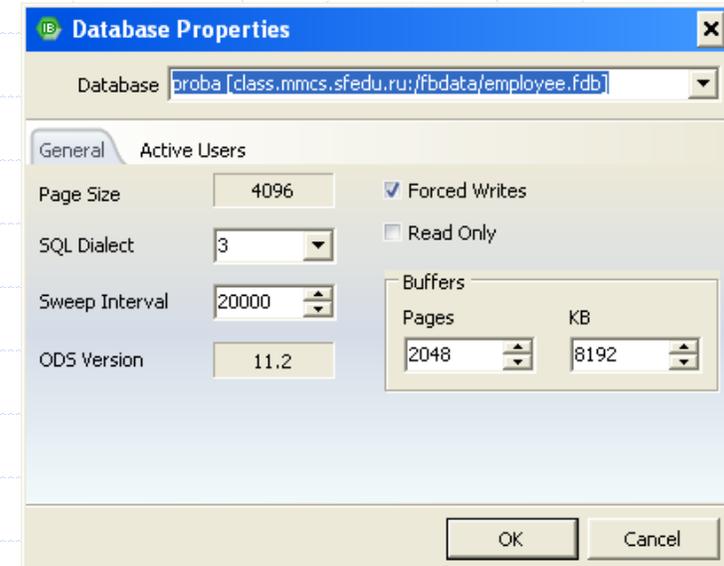
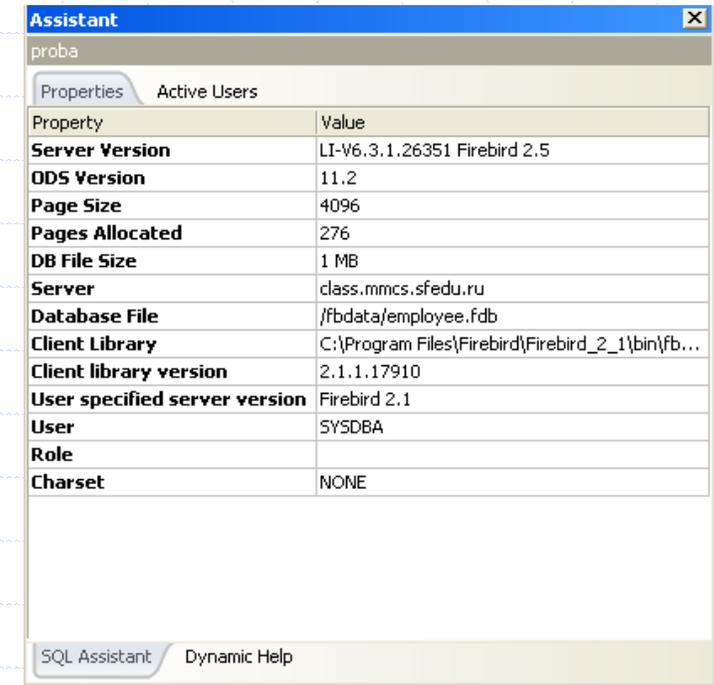
Изменение структуры файлов базы данных, переключение её в состояние «безопасное для копирования» или изменение некоторых свойств базы данных.

База данных должна быть открытой.

```
ALTER {DATABASE | SCHEMA}
    {<add_sec_clause> [<add_sec_clausees> ...]}
| {ADD DIFFERENCE FILE 'diff_file' | DROP DIFFERENCE FILE}
| {{BEGIN | END} BACKUP}
| {SET DEFAULT CHARACTER SET charset}
| {SET LINGER TO seconds | DROP LINGER}
| {ENCRYPT WITH plugin_name [KEY key_name] | DECRYPT};
```

Получение информации

- ◆ isql
 - SHOW DATABASE;
- ◆ IBExpert
 - Database explorer (properties)
 - Services->Database properties



Основные объекты схемы

- ◆ DOMAIN
- ◆ EXCEPTION
- ◆ FUNCTION
- ◆ GENERATOR
- ◆ PROCEDURE
- ◆ ROLE
- ◆ SHADOW
- ◆ TABLE
- ◆ CONSTRAINT Ограничения целостности таблиц (связаны с таблицей)
 - NOT NULL
 - CHECK
 - PRIMARY KEY
 - UNIQUE
 - FOREIGN KEY
- ◆ INDEX (связаны с таблицей)
- ◆ TRIGGER (связаны с таблицей)
- ◆ VIEW (связаны с таблицами)

Создание таблицы

```
CREATE TABLE table [EXTERNAL [FILE] 'filespec']
```

```
(col_def [, col_def | tconstraint ...]);
```

```
col_def = col_name {datatype | COMPUTED [BY] (expr) | domain}
```

```
[DEFAULT {literal | NULL | USER}]
```

```
[NOT NULL]
```

```
[col_constraint]
```

```
[COLLATE collation]
```

Пример

```
CREATE TABLE EMPLOYEE (  
  EMP_NO      EMPNO NOT NULL /* EMPNO = SMALLINT */,  
  ...  
  HIRE_DATE   TIMESTAMP DEFAULT 'NOW' NOT NULL,  
  ...  
  SALARY      NUMERIC(10,2) NOT NULL  
              CHECK (VALUE > 0), /* сложный constraint через JOB*/  
  FULL_NAME   COMPUTED BY (last_name || ', ' || first_name)  
);
```

Table : [NEW_TABLE] : proba (class.mmcs.sfedu.ru:/fbdata/employee.fdb)

Table        Type: Persistent **NEW_TABLE**

External File

Fields Description Comparison To-do

NEW_FIELD INTEGER

#	PK	Field Name	Field Type	Domain	Size	Scale	Subtype	Array	Not Null	Charset	Collate	Desc
		NEW_FIELD	INTEGER						<input type="checkbox"/>			

Field description Field dependencies

Типы данных

{SMALLINT | INTEGER | FLOAT | DOUBLE PRECISION} [array_dim]

| {DATE | TIME | TIMESTAMP} [array_dim]

| {DECIMAL | NUMERIC} (precision [, scale]) [array_dim]

| {CHAR | CHARACTER | CHARACTER VARYING | VARCHAR} (length) [array_dim]

[CHARACTER SET charname]

| {NCHAR | NATIONAL CHARACTER | NATIONAL CHAR} [VARYING] (length) [array_dim]

| BLOB [SUB_TYPE {int | subtype_name}]

[SEGMENT SIZE int]

[CHARACTER SET charname]

| BLOB [(seglen [, subtype])] array_dim = [[x:]y [, [x:]y ...]]

Constraints

- ◆ Средства контроля за корректностью информации в БД
- ◆ Таблица не может содержать ни одной строки, в которой бы нарушалось ограничение, независимо от момента создания ограничения
- ◆ Попытка добавить новое ограничение, если в таблице уже есть данные, нарушающие его, завершится неудачей
- ◆ Ограничения могут создаваться одновременно с созданием таблицы или после создания
- ◆ Добавление/удаление ограничений через команду ALTER TABLE

Ограничения столбца

col_constraint = [CONSTRAINT constraint]

{ UNIQUE | PRIMARY KEY

| REFERENCES other_table [(other_col [, other_col ...])]

[ON DELETE {NO ACTION| CASCADE| SET DEFAULT| SET NULL}]

[ON UPDATE {NO ACTION| CASCADE| SET DEFAULT| SET NULL}]

| CHECK (check_condition)}

NOT NULL можно считать ограничением столбца, требуется задавать сразу при создании таблицы

NULL

- ◆ SQL поддерживает возможность представления отсутствующих данных путем введения признака (NULL–значений)
- ◆ СУБД резервирует в каждой хранимой строке по одному дополнительному биту на каждый столбец, в котором допустимы NULL–значения.
- ◆ Дополнительный бит служит «индикатором» присутствия или отсутствия содержательного значения в самом поле
- ◆ Стандарт ANSI/ISO и большинство СУБД поддерживают возможность задать для отдельных столбцов в качестве ограничения целостности условие обязательности данных (NOT NULL)
- ◆ Как правило, СУБД не позволяют изменять это условие для таблиц с помощью оператора ALTER TABLE

Особенности NULL

- ◆ Большинство СУБД значение арифметического или символьного выражения, среди операндов которого встретилось значение NULL, полагает равным NULL
- ◆ Логическое выражение, среди операндов которого встретилось NULL, принимает значение UNKNOWN – не определено

Всегда unknown !!!

«=NULL» или «<>NULL»

Всегда NULL !!!

« +NULL »

« || NULL »

Предикат проверки

IS NULL

IS NOT NULL

SQL-92

$\text{NULLIF}(v1,v2) = \begin{cases} \text{NULL}, & v1=v2 \\ v1, & \text{иначе} \end{cases}$

```
update RESERV set KOL=NULLIF(KOL,-1);
```

$\text{COALESCE}(v1,v2,v3,\dots,vn)$ – первое NOT NULL из списка или NULL

```
select NAME, COALESCE (STATE,0.0) + COALESCE(COMIS,0.0)
from PAY;
```

- ◆ Значения NULL недопустимы в столбцах, входящих в первичный ключ
- ◆ NULL и UNIQUE
- ◆ Если предварительный анализ показывает, что в большинстве строк постоянно будет присутствовать значение NULL, лучше выделить такой столбец в отдельную таблицу, установив связь с исходной с помощью внешнего ключа

Ограничения таблицы

Задаются в конце определения структуры таблицы

tconstraint = [CONSTRAINT constraint]

{**PRIMARY KEY** | **UNIQUE**} (col [, col ...])

| **FOREIGN KEY** (col [, col ...]) REFERENCES other_table

[ON DELETE {NO ACTION | CASCADE | SET DEFAULT | SET NULL}]

[ON UPDATE {NO ACTION | CASCADE | SET DEFAULT | SET NULL}]

| **CHECK** (check_condition)}

- 
- ◆ Добавление/удаление ограничений через команду ALTER TABLE
 - ◆ Таблица не может содержать ни одной строки, в которой бы нарушалось ограничение
 - ◆ Попытка добавить новое ограничение, если в таблице уже есть данные, нарушающие его, завершится неудачей

Ограничение CHECK

- ◆ Если ограничение на допустимые значения может быть сформулировано в виде условия, его можно задать с помощью элемента базы данных – ограничения (constraint) CHECK
- ◆ Ограничения CHECK могут быть, как определены сразу при создании таблицы в операторе CREATE TABLE, так и добавлены (изменены или удалены) в операторе ALTER TABLE
- ◆ Ограничения CHECK удобно задавать в доменах, создавая пользовательский тип данных с контролем значений

Ограничение CHECK

```
[CONSTRAINT name_constr] CHECK (check_condition)
```

Пример

```
CREATE TABLE ATABLE (ACOLUMN ...  
...CONSTRAINT ACOLUMN_CH  
CHECK (ACOLUMN > 10),  
...  
)
```

Ограничение CHECK

- ◆ Синтаксис ограничения CHECK не требует, чтобы условие было сформулировано только относительно значения одного столбца.
- ◆ В условие проверки могут быть включены значения из других столбцов, других строк и даже других таблиц.
- ◆ Столбец может иметь только одно ограничение CHECK, хотя его логика может быть сколь угодно сложной
- ◆ Ограничение CHECK для столбца, основанного на домене, не может изменять наследуемую от домена проверку, она может быть только усилена

Пример

```
salary >= (SELECT min_salary FROM job WHERE  
            job.job_code = employee.job_code AND  
            job.job_grade = employee.job_grade AND  
            job.job_country = employee.job_country)
```

AND

```
salary <= (SELECT max_salary FROM job WHERE  
            job.job_code = employee.job_code AND  
            job.job_grade = employee.job_grade AND  
            job.job_country = employee.job_country)
```

Ограничение первичного ключа

```
create table PERSON(  
    PERSON_ID integer NOT NULL PRIMARY KEY,  
    ...  
);
```

```
create table ORDER (    ID_WH integer NOT NULL,  
                        DATA_ORD date NOT NULL,  
                        SUM numeric(15,2),  
                        CONSTRAINT PK_ORDER PRIMARY KEY(ID_WH,DATA_ORD));
```

Лучше определять в команде ALTER

```
alter table BOOK  
    add constraint PK_BOOK PRIMARY KEY (ID_BOOK);
```

Primary key

Table : [EMPLOYEE] : empl_local (localhost:C:\Program Files\Firebird\Firebird_2_5\examples\empbuild\EMPL...)

Table | ⚡ | ✓ | ✗ | 📅 | 🖨️ | 🌸 | 📊 | 🗑️ | Get record count | EMPLOYEE

Fields | **Constraints** | Indices | Dependencies | Triggers | Data | Master/Detail View | Description | DDL | Grants

1.Primary key | 2.Foreign keys | 3.Checks | 4.Uniques

Constraint Name	On Field	Index Name	Index Sorting
INTEG_27	EMP_NO	RDB\$PRIMARY7	Ascending

PK	Field Name	Field Type	Domain	Size	Scale	Subtype	Array	Not Null	Charset	Collate	Descript...	AI
1	ID_T1	INTEGER						<input checked="" type="checkbox"/>				<input type="checkbox"/>

Autoincrement Field

Generator | Trigger | Procedure

Internal generator

Use internal sequence generator (identity column)

User defined generator

Create Generator

Use existing generator

Требования ограничения первичного ключа

- ◆ Все столбцы, входящие в первичный ключ должны иметь ограничение NOT NULL
- ◆ При определении ограничения PRIMARY KEY для комбинации значений, входящих в первичный ключ автоматически устанавливается ограничение целостности UNIQUE
- ◆ Кроме того, для каждого первичного ключа автоматически создается индекс, ускоряющий поиск по ключу.
- ◆ Ограничение первичного ключа командой ALTER TABLE будет создано только в том случае, если содержащиеся в таблице данные не противоречат этому ограничению.

Выбор первичного ключа

- ◆ Хотя само ограничение PRIMARY KEY не является ссылочным ограничением, оно обычно является частью ограничения ссылочной целостности
- ◆ Рекомендуется при реализации базы данных заменять составные первичные ключи и ключи, основанные на строковых типах, простыми числовыми ключами
- ◆ Для этого достаточно ввести в таблицу дополнительное целочисленное поле, имеющее смысл: «порядковый номер», «идентификационный номер» или «регистрационный номер»
- ◆ Такой первичный ключ называется *суррогатным*

Суррогатный первичный ключ

- ◆ Для упрощения работы с суррогатными первичными ключами, пользователь обычно не должен заниматься вводом их значений. Вместо этого используются автоинкрементные типы данных или заменяющие их механизмы (последовательности, генераторы, триггеры).

PK	Field Name	Field Type	Domain	Size	Scale	Subtype	Array	Not Null	Charset	Collate	Descript...	AI
1	ID_T1	INTEGER						<input checked="" type="checkbox"/>				<input type="checkbox"/>

Autoincrement Field

Generator Trigger Procedure

Internal generator

Use internal sequence generator (identity column)

User defined generator

Create Generator

Use existing generator

Ограничение UNIQUE

- ◆ Определяет ограничение возможного ключа
- ◆ Контролирует отсутствие дубликатов
- ◆ NULL
 - допускает одно или более значений NULL в столбце, на который наложено ограничение UNIQUE
 - Для уникальных ключей, содержащих несколько столбцов

```
RECREATE TABLE t( x INT, y INT, z INT, UNIQUE(x,y,z) );
INSERT INTO t VALUES ( NULL, 1, 1 );
INSERT INTO t VALUES ( NULL, NULL, 1 );
INSERT INTO t VALUES ( NULL, NULL, NULL );
INSERT INTO t VALUES ( NULL, NULL, NULL ); -- Разрешено
INSERT INTO t VALUES ( NULL, NULL, 1 );    -- Запрещено
```

Внешний ключ

- ◆ Внешний ключ – это столбец или набор столбцов в одной таблице, которые соответствуют столбцу или набору столбцов определяющих возможный ключ (первичный или unique) другой таблицы
- ◆ Внешний ключ необходим для реализации ссылочной целостности в БД
- ◆ В общем случае внешний ключ реализует необязательную связь один–ко–многим. При этом таблица, на которую ссылается другая таблица, называется родительской. Ссылающаяся таблица называется дочерней
- ◆ Термин «ссылочная целостность» обозначает возможности базы данных защищать себя от получения данных, которые могут послужить источником нарушения связи между двумя таблицами
- ◆ В системах управления реляционными базами данных ссылочная целостность поддерживается посредством ограничения внешнего ключа FOREIGN KEY

Операции, которые могут нарушить ссылочную целостность

- ◆ Добавление новой строки в дочернюю таблицу
- ◆ Изменение строки дочерней таблицы, затрагивающее внешний ключ
- ◆ Изменение строки родительской таблицы, затрагивающее возможный ключ, для которого есть связи из дочерней таблицы
- ◆ Удаление строки родительской таблицы, для которой есть связанные строки в дочерней таблицы

Добавление в дочернюю таблицу

The screenshot shows a database application window with a data table and an error dialog box. The table has the following data:

FISCAL_Y...	PROJ_ID	DEPT_...	QUART_HEAD_CNT	PROJECTED_BUDGET
2 022	DB	621	<null>	100 000,00

The error dialog box contains the following text:

Error Message:

can't format message 13:470 -- message file D:\БД\firebird.msg not found.
violation of FOREIGN KEY constraint "INTEG_48" on table "PROJ_DEPT_BUDGET".
Foreign key reference target does not exist.
Problematic key value is ("PROJ_ID" = 'DB').

[00542283] FIB.IBError (Line 556, "FIB.pas" + 55) + \$2

Buttons: Copy, Close

Удаление из родительской таблицы

The screenshot shows a database management tool interface. At the top, there are tabs for 'Edit: 6', 'History', 'Plan Analyzer', 'Performance Analysis', 'Logs', and 'Find query'. Below these tabs, a SQL query is entered in a text area: `delete from goods where id_goods='T1'`. The value 'T1' is highlighted in cyan. Below the query area, there is a 'Messages' tab with a close button (x) on the left. The 'Messages' tab is active and displays the following error message:
can't format message 13:470 -- message file D:\work\BD\3.0\firebird.msg not found.
violation of FOREIGN KEY constraint "FK_GOODS_WH_2" on table "GOODS_WH".
Foreign key references are present for the record.
Problematic key value is ("ID_GOODS" = 'T1').
An orange hand-drawn circle highlights the error message text.

Определение ограничения целостности внешнего ключа

```
. . .  
[CONSTRAINT имя]  
FOREIGN KEY (столбец [,столбец...])  
REFERENCES таблица (столбец [,столбец...])  
[ON DELETE {NO ACTION | CASCADE |  
            SET NULL | SET DEFAULT}]  
[ON UPDATE {NO ACTION | CASCADE |  
            SET NULL | SET DEFAULT}]
```

в стандарте SQL вместо NO ACTION – RESTRICT

Определение ограничения целостности внешнего ключа

◆ Пример

```
ALTER TABLE OPERATION  
ADD CONSTRAINT FK_OPERATION_4  
FOREIGN KEY (ID_WH)  
REFERENCES WAREHOUSE (ID_WH) ;
```

Определение ограничения целостности внешнего ключа

- ◆ для внешнего ключа создается индекс

PK	Index	On field	Expression	J...	\...	Sorting	St...
	BUDGETX	BUDGET		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Desc...	0,...
	RDB\$4	DEPARTMENT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Asce...	0
	RDB\$FOR...	MNGR_NO		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Asce...	0
	RDB\$FOR...	HEAD_DEPT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Asce...	0
	RDB\$PRI...	DEPT_NO		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Asce...	0

Определение ограничения целостности внешнего ключа

- ◆ Часто создание индекса является нежелательным, но нет возможности удалить индекс пока существует внешний ключ
- ◆ Поэтому в тех случаях, когда наличие индекса для внешнего ключа имеет негативные последствия, для реализации ссылочной целостности можно использовать другие средства, например, триггеры или процедуры

Изменение/удаление таблицы

ALTER TABLE table

[ADD {column | constraint}]|

[DROP {column | constraint}] |

[ALTER {column | constraint}]

DROP TABLE table

Представления

```
CREATE OR ALTER  
  VIEW view [ (col [,col, ...])  
AS select_operator
```

```
DROP VIEW view
```

Домены

- ◆ Назначение доменов – вынести некоторые ограничения целостности над базовым типом данных СУБД на уровень описания схемы базы данных.
- ◆ Все столбцы в разных таблицах, которые были определены с одним доменом, будут иметь идентичный тип данных и идентичные ограничения, за исключением тех ограничений, которые будут переопределены на уровне описания таблицы.

Домены

```
CREATE DOMAIN имя домена  
  [AS] тип данных  
  [DEFAULT литерал]  
  [NOT NULL]  
  [CHECK (условие)];
```

Домены

```
create domain CUSTNO  
as integer  
default 9999  
check (VALUE>1000);
```

```
create domain TELNO  
as integer  
check ((value is null) or  
      (value between 100 and 999));
```

Домены

ALTER DOMAIN . . .

Нельзя добавить или убрать NOT NULL

DROP DOMAIN . . .

Использование доменов

```
create table TC  
(id_cust CUSTNO NOT NULL,  
  cust_tel TELNO,  
  ...)
```

NOT NULL можно только добавить
DEFAULT можно изменить
CHECK можно только усилить

Что еще должно быть определено в схеме БД?

- ◆ Индексы для полей (комбинаций полей), по которым часто идет поиск
- ◆ Хранимые процедуры
- ◆ Триггеры
- ◆ Генераторы
- ◆ Исключения
- ◆ Пользователи и роли