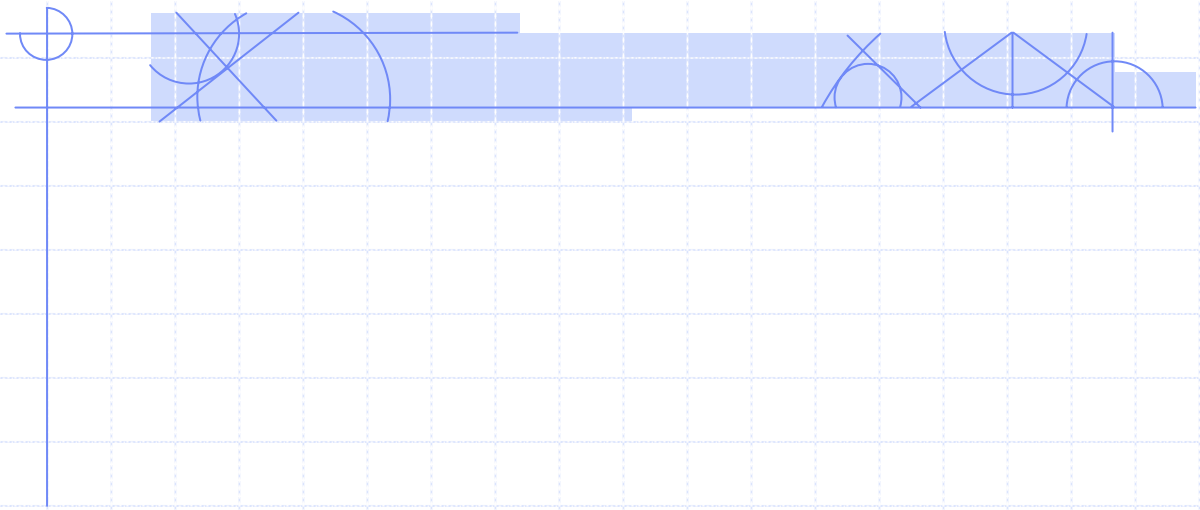
SQL



SQL

SQL structured query language

SQL = DDL + DML + DCL

- DDL data description language
- DML data manipulation language
- DCL data control language

DDL

Дефиниране на таблици, атрибути, отношения и индекси

CREATE TABLE	създава нова таблица
ALTER TABLE	променя структурата на съществуваща таблица
DROP TABLE	изтрива таблица
CREATE INDEX	създава индекс за поле или група полета
DROP INDEX	изтрива индекс

DML

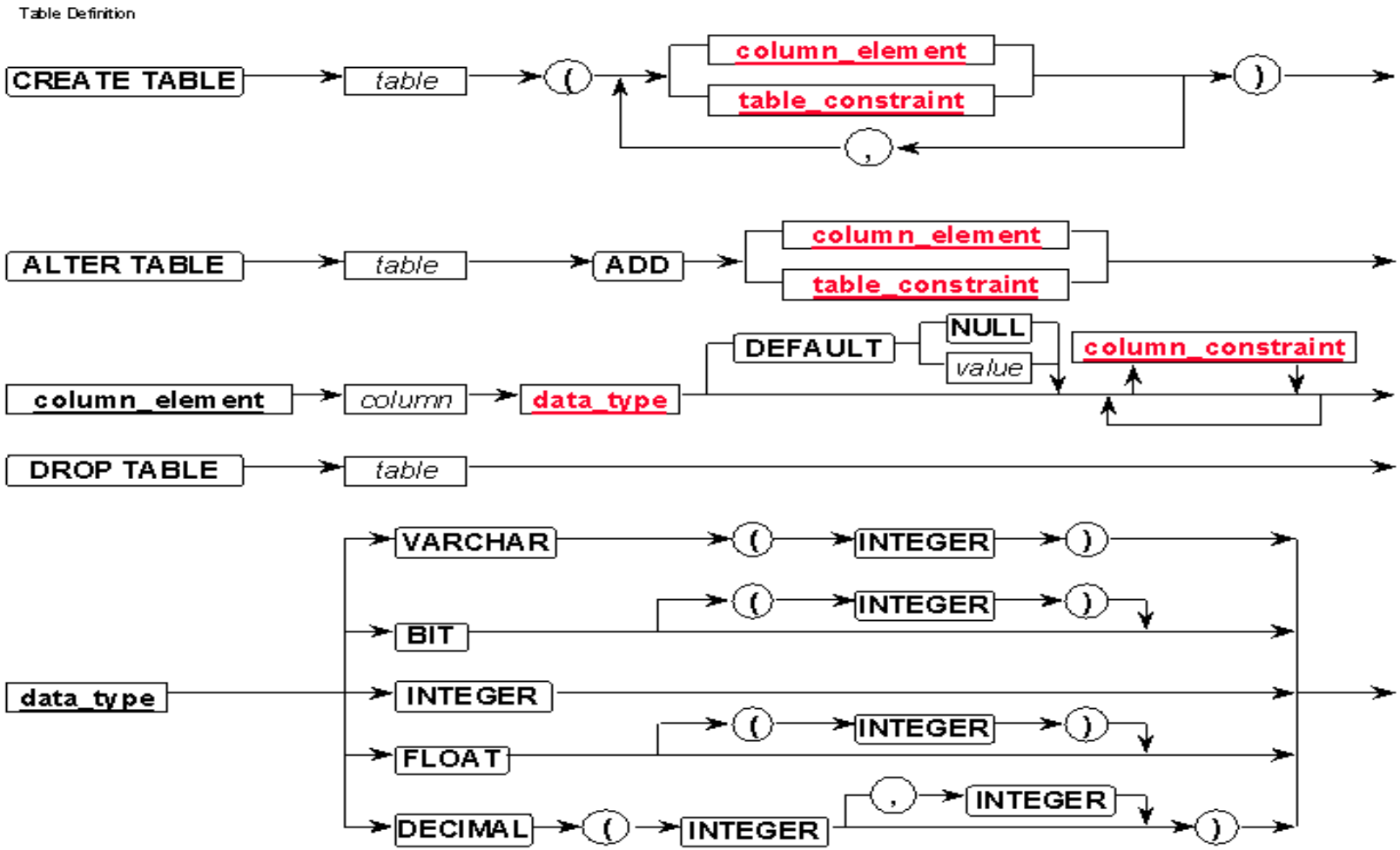
Извличане, вмъкване и изменяне на съдържанието на базата данни

INSERT INTO	добавя записи в таблица
DELETE FROM	изтрива записи от таблица
UPDATE	актуализира записи
SELECT ... INTO	записва резултати в таблица
SELECT... FROM ... WHERE ...	извлича информация

SQL: Синтактични диаграми

- Операции с таблици
- Обработка на данни
- Операции със заявки
- Търсене
- Изрази
- Ограничения
- Визуализация

Операции с таблици



create table

- Формат:

```
create table "tablename"
  ("column1" "data type",
   "column2" "data type",
   "column3" "data type");
create table "tablename"
  ("column1" "data type" [constraint],
   "column2" "data type" [constraint],
   "column3" "data type" [constraint]);
```

create table

- Пример:

```
create table employee
  (first varchar(15),
   last varchar(20),
   age number(3),
   address varchar(30),
   city varchar(20),
   state varchar(20));
```

create table

- Основни типове данни

char(size) Стринг с фиксирана дължина, max 255 байта

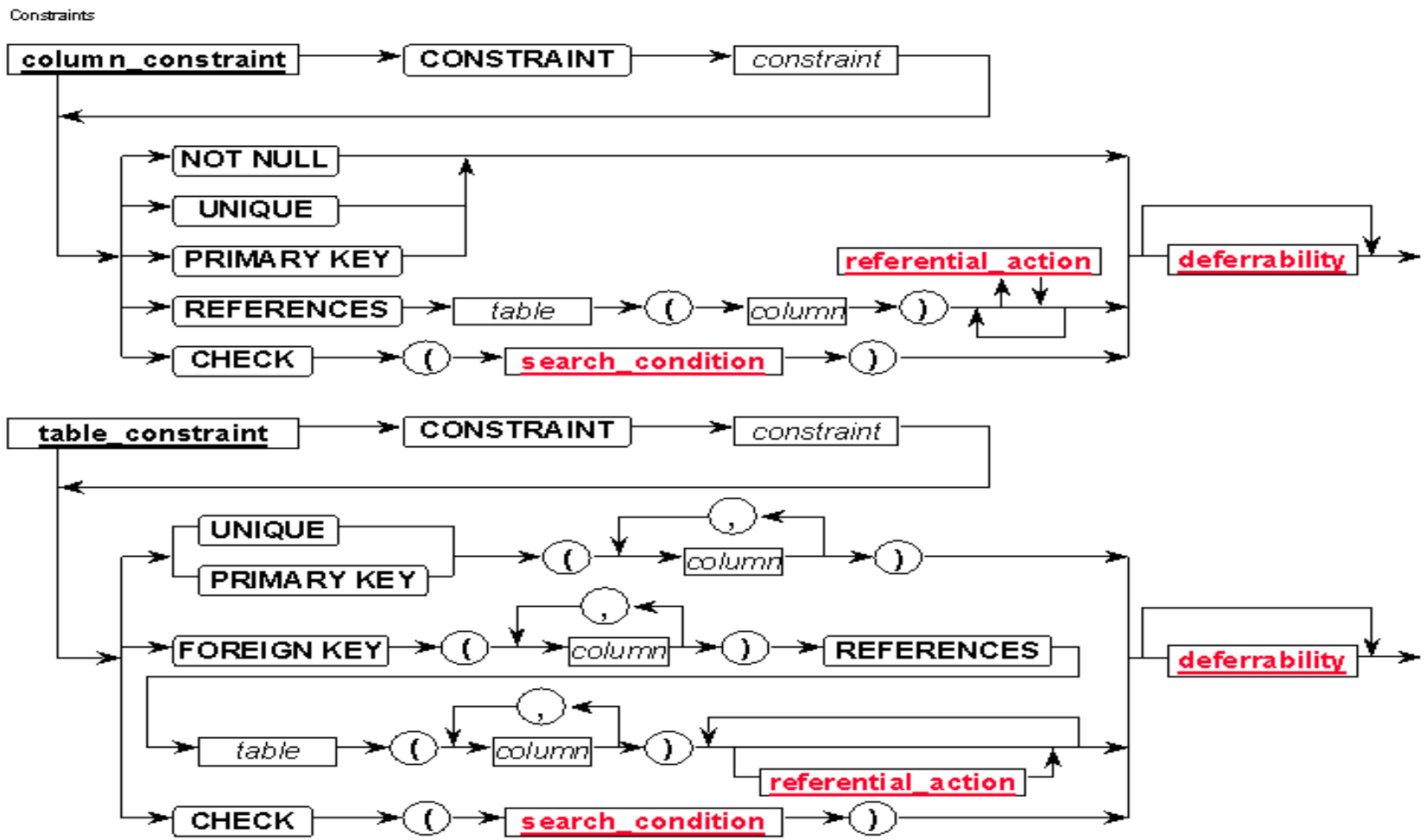
varchar(size) Стринг с променлива дължина, max size – в скоби

number(size) Числова стойност с max брой цифри

date Дата

number(size,d) Числова стойност с max брой цифри общо "size" и след десетичната точка цифри "d"

Ограничения



constrains

- Правило, асоциирано с колона, което данните в колоната трябва да спазват
- Видове
  - "unique" – не може два или повече записи да имат еднакви стойности в тази колона
  - "not null" – не може да има празна клетка в тази колона
  - "primary key" - идентификатор на реда, първичен ключ

drop table

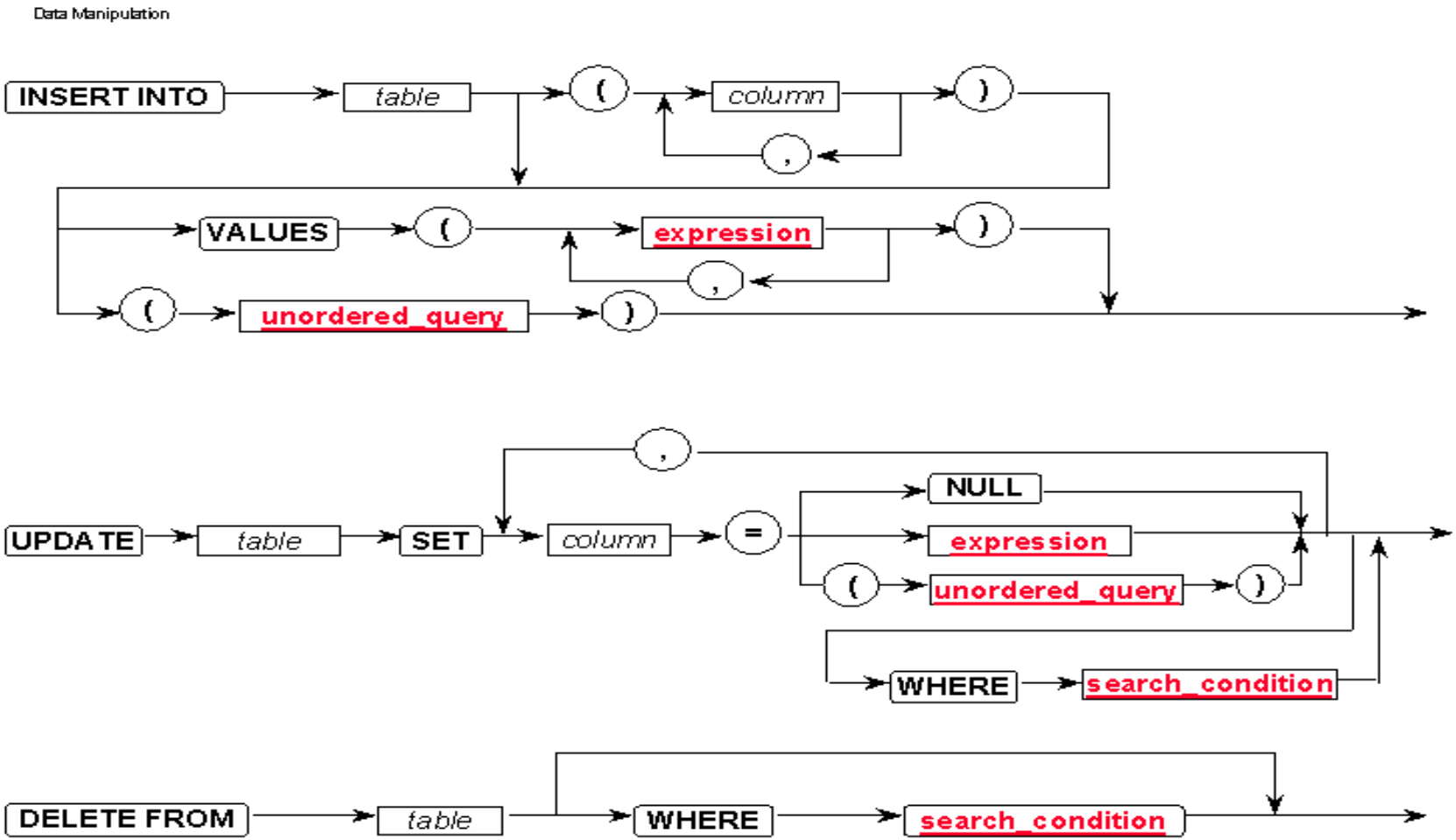
- Модел:

```
drop table "tablename"
```

- Пример:

```
drop table myemployees;
```

Обработка на данни



insert into

Модел:

```
insert into "tablename"
  (first_column,...last_column) values
  (first_value,...last_value);
```

Пример:

```
insert into employee (first, last, age,
  address, city, state) values ('Luke','Duke',
  45, '2130 Boars Nest', 'Hazard Co',
  'Georgia');
```

delete from

- Модел:

```
delete from "tablename"
  where "columnname" OPERATOR "value" [and|or
  "column" OPERATOR "value"];
```

- Примери:

```
delete from employee;
delete from employee where lastname = 'May';
delete from employee where firstname = 'Mike'
  or firstname = 'Eric';
```

update

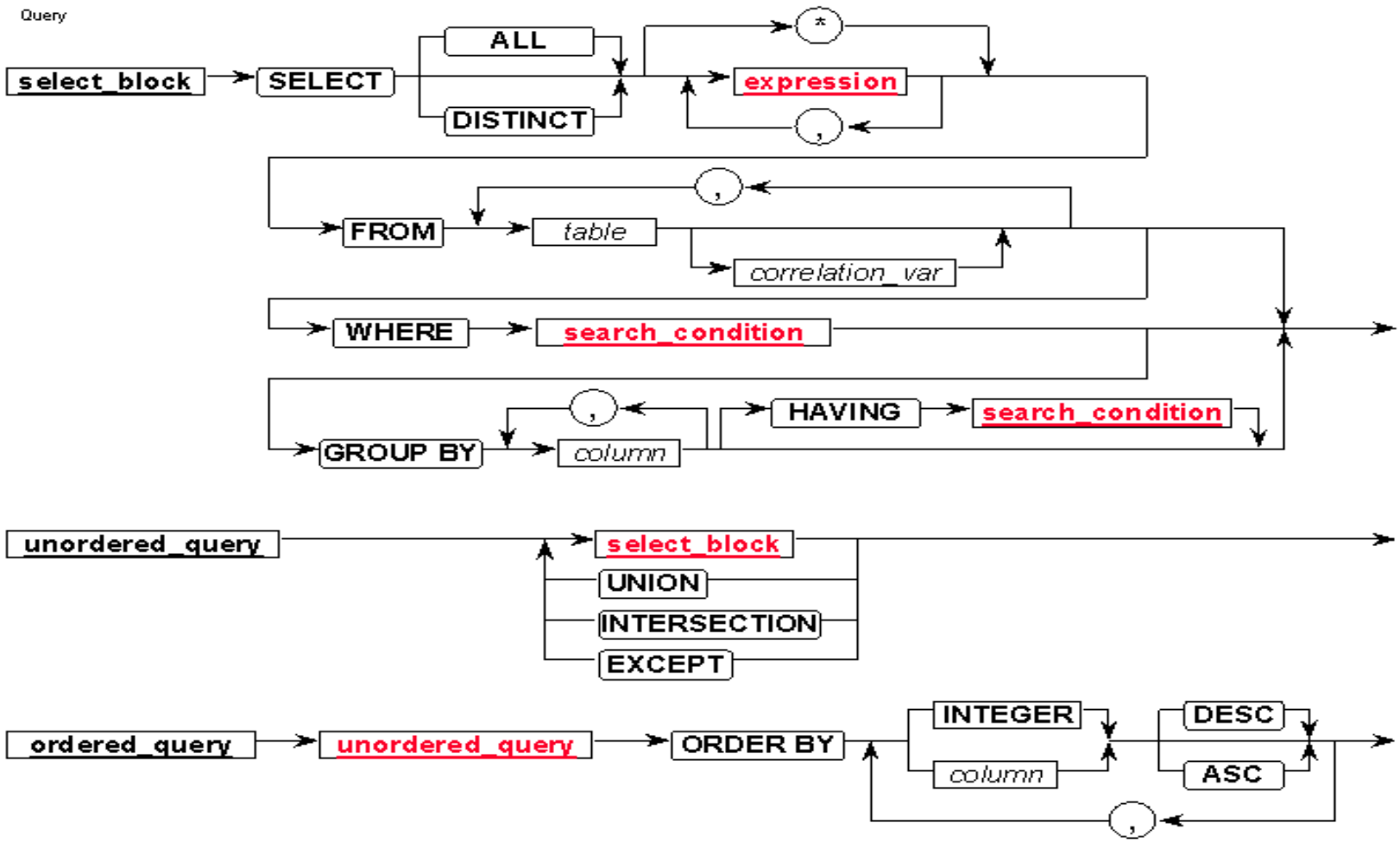
- Модел:

```
update "tablename" set "columnname" =
  "newvalue" [,"nextcolumn" = "newvalue2"...]
  where "columnname" OPERATOR "value" [and|or
  "column" OPERATOR "value"]; [] = optional
```

- Пример:

```
update phone_book set area_code = 623 where
  prefix = 979;
update phone_book set last_name = 'Smith',
  prefix=555, suffix=9292 where last_name =
  'Jones';
update employee set age = age+1 where
  first_name='Mary' and last_name='Williams';
```

Операции със заявки



select

- Модел:

```
select "column1" [,"column2",etc] from
  "tablename" [where "condition"];
```

- Примери:

```
select first, last, city from empinfo where
  first LIKE 'Er%';
select first, last from empinfo where last
  LIKE '%s';
select * from empinfo where first = 'Eric';
```

select

- Формат:

```
SELECT [ALL | DISTINCT] column1[,column2]
  FROM table1[,table2]
  [WHERE "conditions"]
  [GROUP BY "column-list"]
  [HAVING "conditions]
  [ORDER BY "column-list" [ASC | DESC] ]
```

Агрегиране

- Функции:
  - MIN – връща най-малката стойност в колоната
  - MAX - връща най-голямата стойност в колоната
  - SUM - връща сумата от числените стойности в колоната
  - AVG - връща средната стойност в колоната
  - COUNT - връща броя стойности в колоната
  - COUNT(*) – връща броя редове в таблицата

- Примери:

```
SELECT AVG(salary) FROM employee;
SELECT AVG(salary) FROM employee WHERE
  title = 'Programmer';
SELECT Count(*) FROM employees;
```

group by

- Агрегиране върху група редове

- Примери:

```
SELECT column1, SUM(column2)
  FROM "list-of-tables"
  GROUP BY "column-list";

SELECT max(salary), dept
  FROM employee
  GROUP BY dept;
```

having

- Условие за участие в group
- Модел:

```
SELECT column1, SUM(column2)
  FROM "list-of-tables"
  GROUP BY "column-list"
  HAVING "condition";
```

- Пример:

```
SELECT dept, avg(salary) FROM employee
  GROUP BY dept HAVING avg(salary) > 20000;
```

order by

- Модел:

```
SELECT column1, SUM(column2)
  FROM "list-of-tables"
  ORDER BY "column-list" [ASC | DESC];
```

- Примери:

```
SELECT employee_id, dept, name, age, salary
  FROM employee_info
  WHERE dept = 'Sales'
  ORDER BY salary;

SELECT employee_id, dept, name, age, salary
  FROM employee_info
  WHERE dept = 'Sales'
  ORDER BY salary, age DESC;
```

join

- Търсене в две или повече таблици едновременно
- Модел:

```
SELECT "list-of-columns"
   FROM table1 JOIN table2
   ON key1 = key2 WHERE "search-condition(s)"
```

- Примери:

```
SELECT title, artist FROM album JOIN track
ON (album.asin=track.album)WHERE song = 'Alison';

SELECT title, COUNT(*)
  FROM album JOIN track ON (asin=album)
 GROUP BY title
```

join

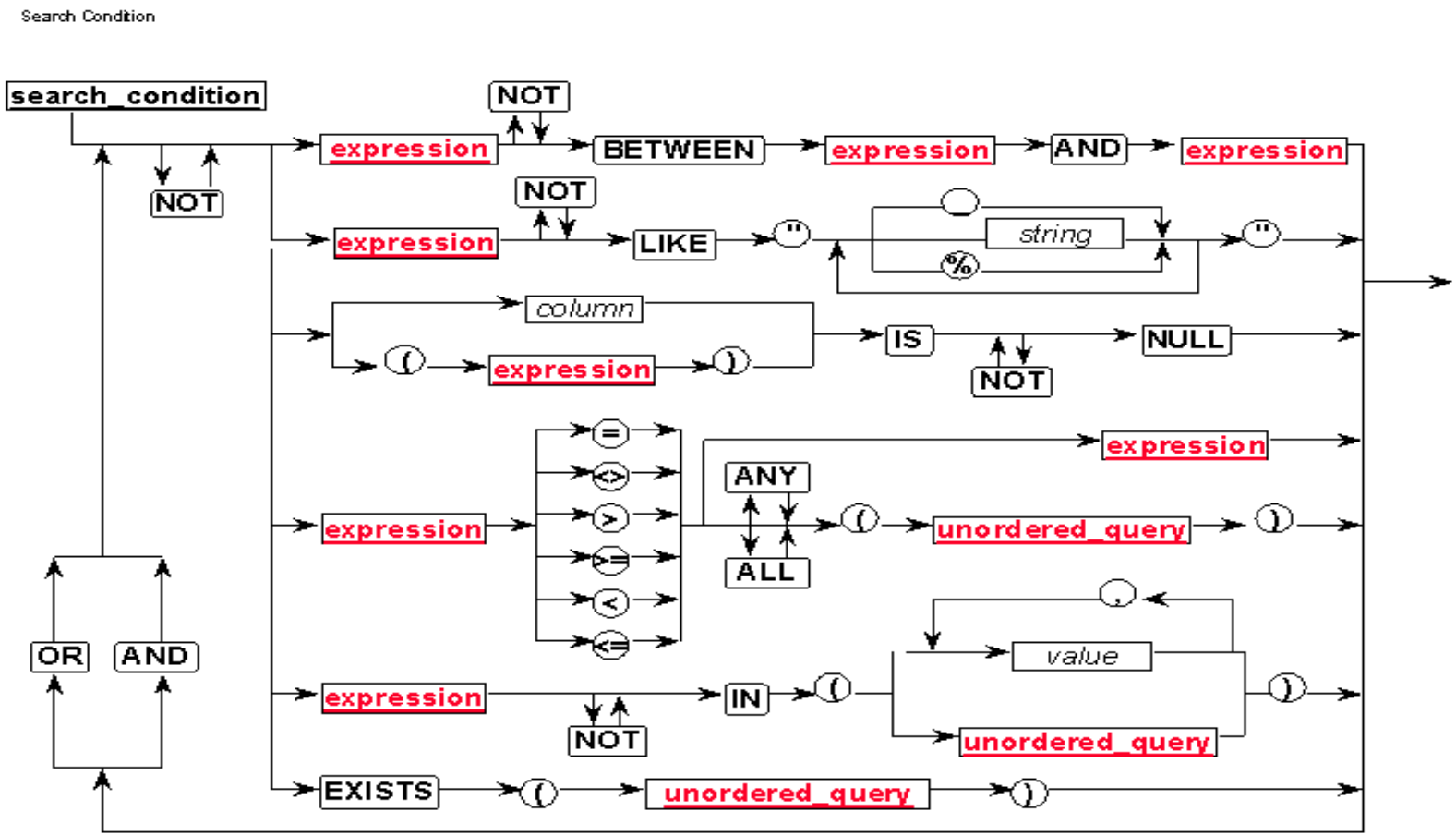
- Модел:

```
SELECT "list-of-columns"
  FROM table1,table2
  WHERE "search-condition(s)"
```

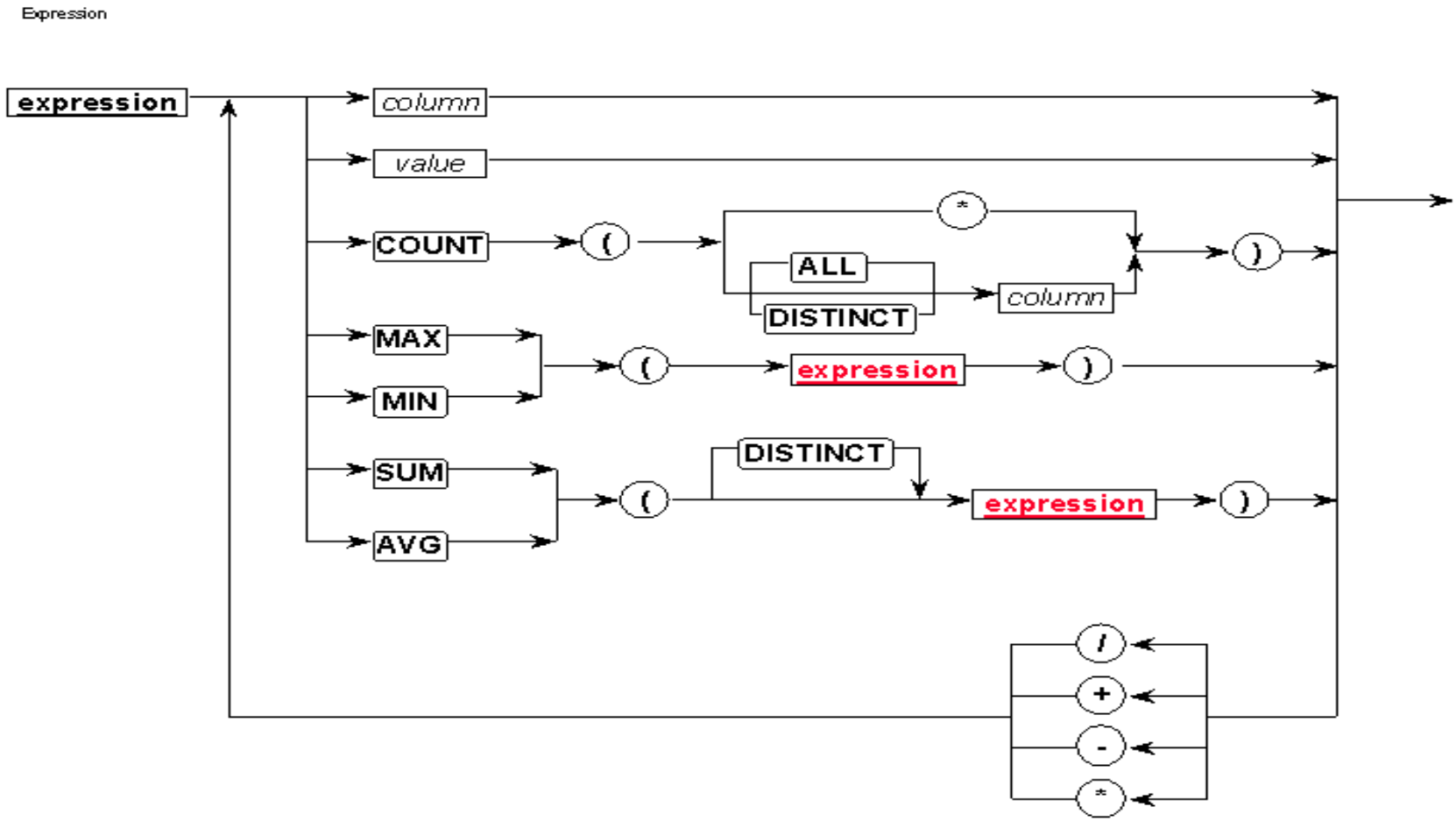
- Пример:

```
SELECT customer_info.firstname,
  customer_info.lastname,purchases.item
  FROM customer_info, purchases
  WHERE customer_info.customer_number =
  purchases.customer_number;
```

Търсене


Search Condition
search_condition
NOT
expression
BETWEEN
AND
LIKE
string
column
IS
NULL
ANY
ALL
unordered_query
value
IN
EXISTS
OR

Изрази


Expression
expression
column
value
COUNT
(
*
ALL
DISTINCT
column
)
MAX
MIN
(
expression
)
SUM
AVG
(
DISTINCT
expression
)
/
+
-
*

Оператори в изрази

- = Equal
- > Greater than
- < Less than
- >= Greater than or equal
- <= Less than or equal
- <> Not equal to
- LIKE *

Оператори в изрази

- in
  - Модел:

```
SELECT col1, SUM(col2)
  FROM "list-of-tables"
  WHERE col3 IN (list-of-values);
```

  - Пример:

```
SELECT employeeid, lastname, salary
  FROM employee_info
  WHERE lastname IN ('Hernandez', 'Jones',
  'Roberts', 'Ruiz');
```

Оператори в изрази

- between
  - Модел:

```
SELECT col1, SUM(col2)
  FROM "list-of-tables"
  WHERE col3 BETWEEN value1 AND value2;
```

  - Примери:

```
SELECT employeeid, age, lastname, salary
  FROM employee_info
  WHERE age BETWEEN 30 AND 40;
```

Оператори в изрази

- Математически оператори
  + addition
  - subtraction
  * multiplication
  / division
  % modulo

- Математически функции
  ABS(x) - връща |x|
  SIGN(x) – връща знака на x as -1, 0, or 1
  MOD(x,y) modulo – остатъкът от делението на x и y (x%y)
  FLOOR(x) – най-голямото цяло <= x
  CEILING(x) или CEIL(x) - най-малкото цяло >= x
  POWER(x,y) - x на степен y
  ROUND(x) – закръгление на x до най-близкото цяло
  ROUND(x,d) – закръгление на x до знак d
  SQRT(x) – квадратен корен от x

Потребители

- Зависи от СУБД
- Нов потребител

```
sp_addlogin 'scott', 'tiger';
CREATE DATABASE scott;
USE scott;
sp_changedbowner scott
```

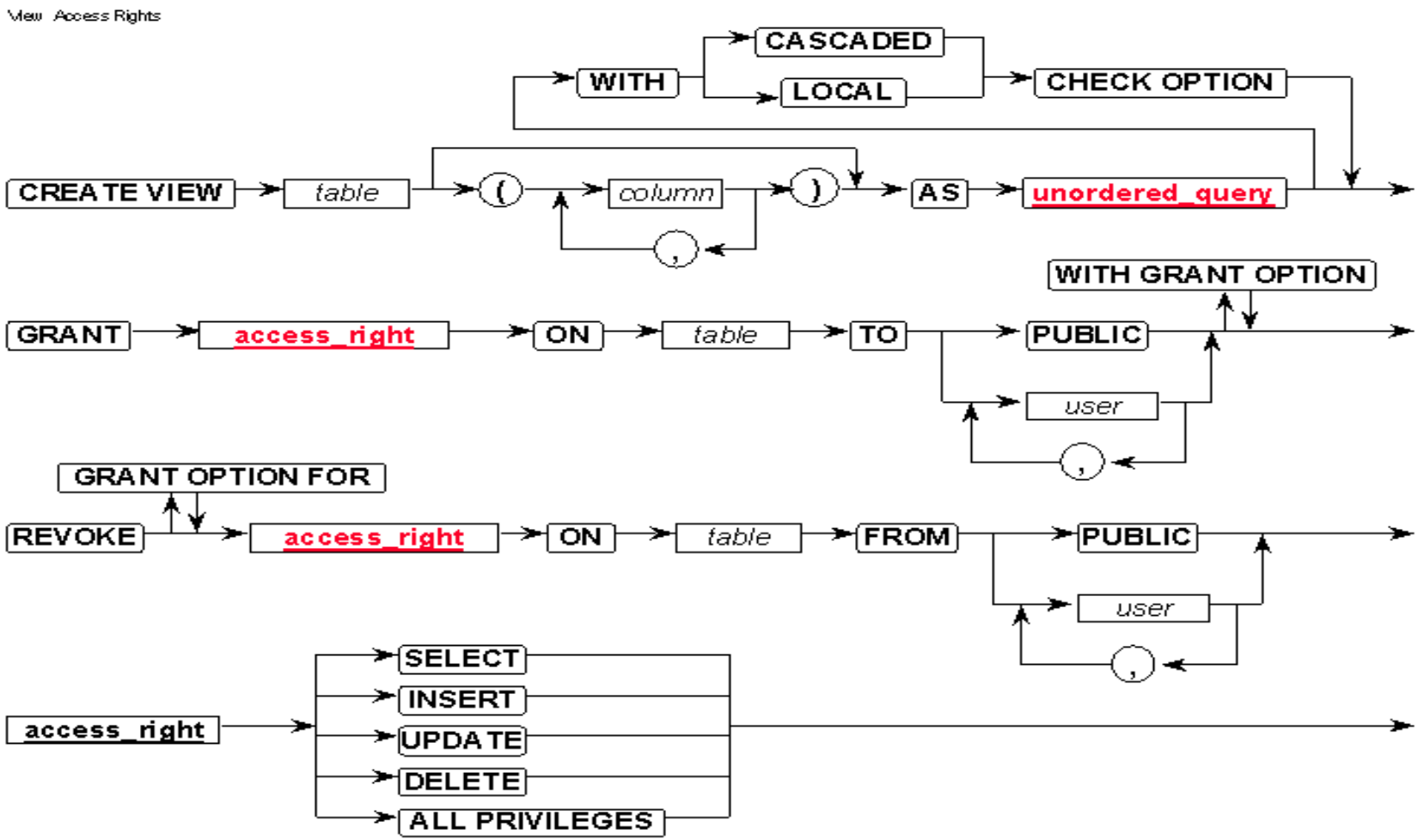
- Смяна на парола

```
sp_password @old='tiger', @new='tiger'
```

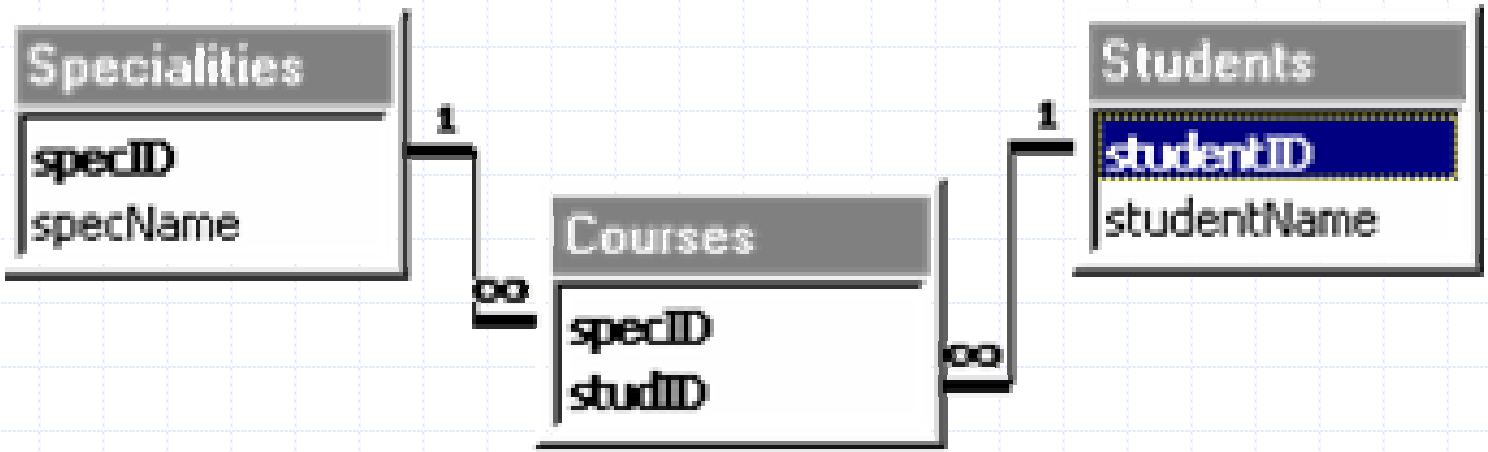
- Идентификация на потребител и собственик (dbo)

```
USE scott;
SELECT user, user_id();
USE gisq;
SELECT user, user_id()
```

Визуализация



Още примери

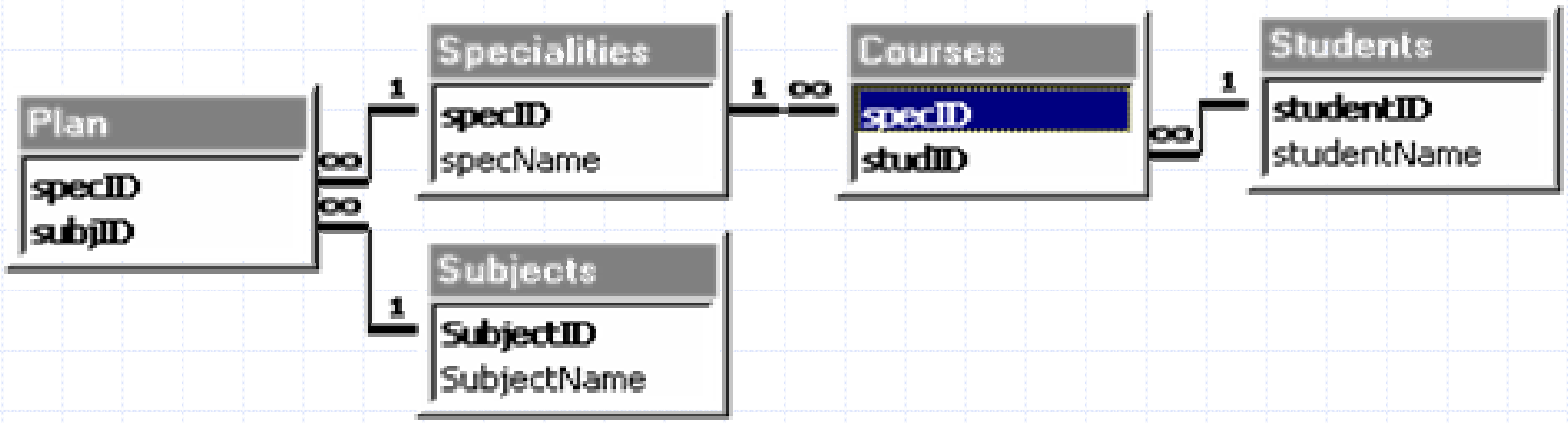


Specialities (specID, specName)

Students (studentID, studentName)

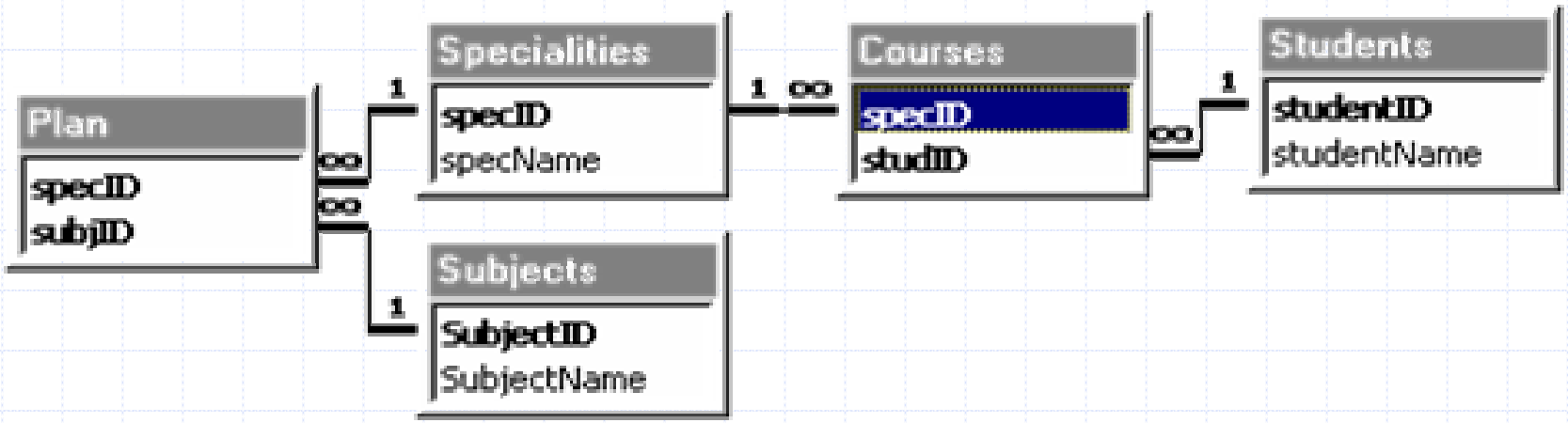
Courses (specID, studID)

Примери




```
CREATE TABLE Plan
(
    specID integer,
    subjID integer,
    constraint tc1 PRIMARY KEY (specID, subjID),
    constraint tc2 FOREIGN KEY (specID) references Specialities,
    constraint tc3 FOREIGN KEY (subjID) references Subjects
);
```

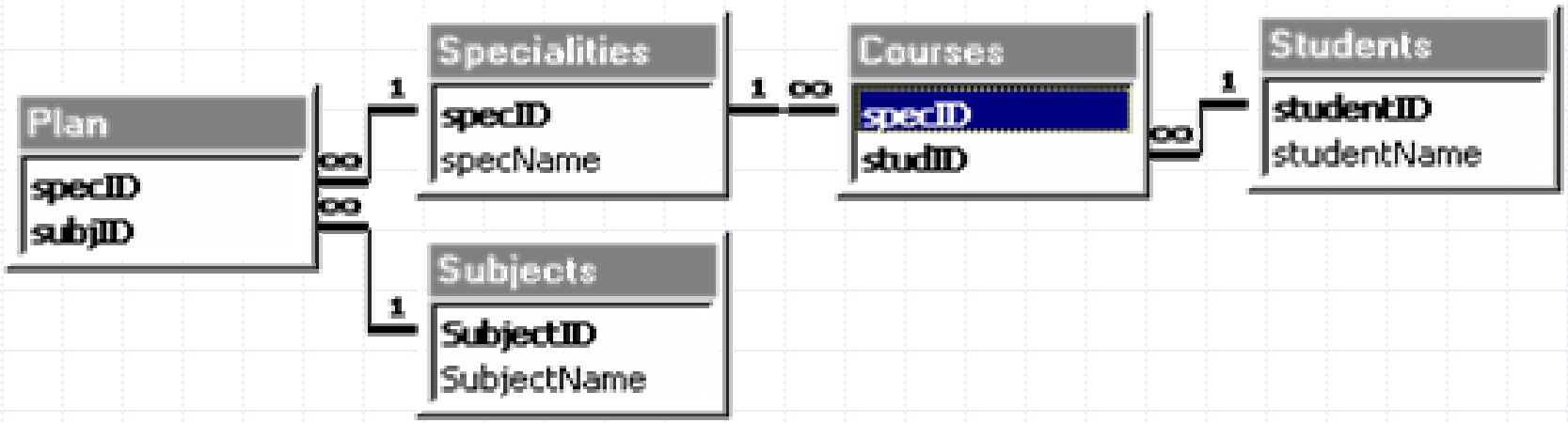
Примери



INSERT INTO TABLE Plan (specID, subjID) VALUES (2, 2);

INSERT INTO TABLE Plan (specID 2, subjID 4);

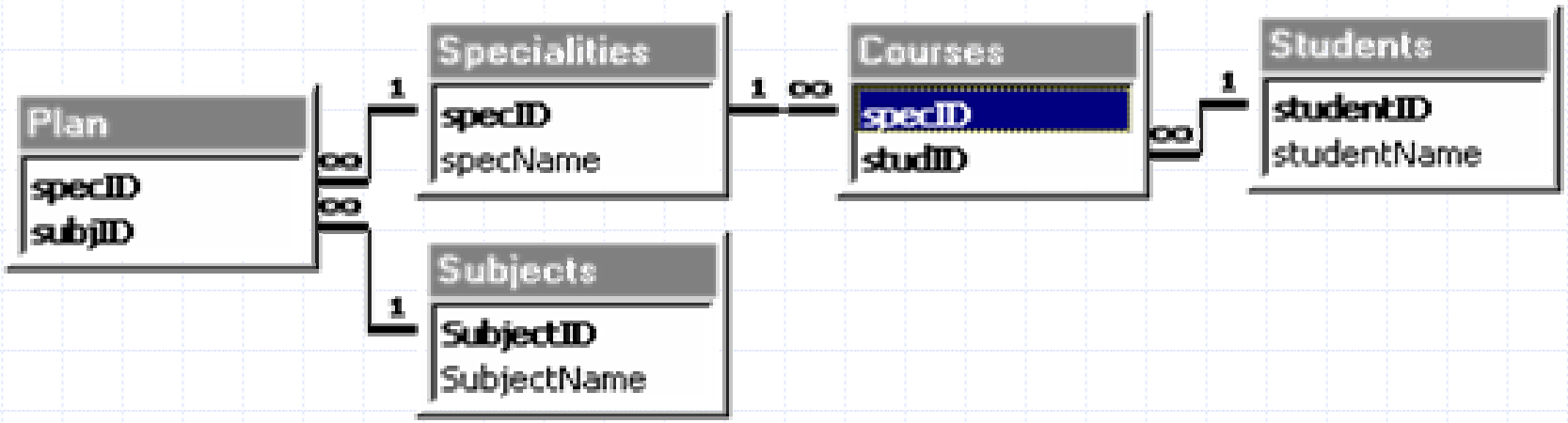
Примери




```
DELETE FROM Students WHERE studentName = 'Maria';

UPDATE Students
    SET studentID = studentID + 1
    WHERE studentName = 'Kate';
```

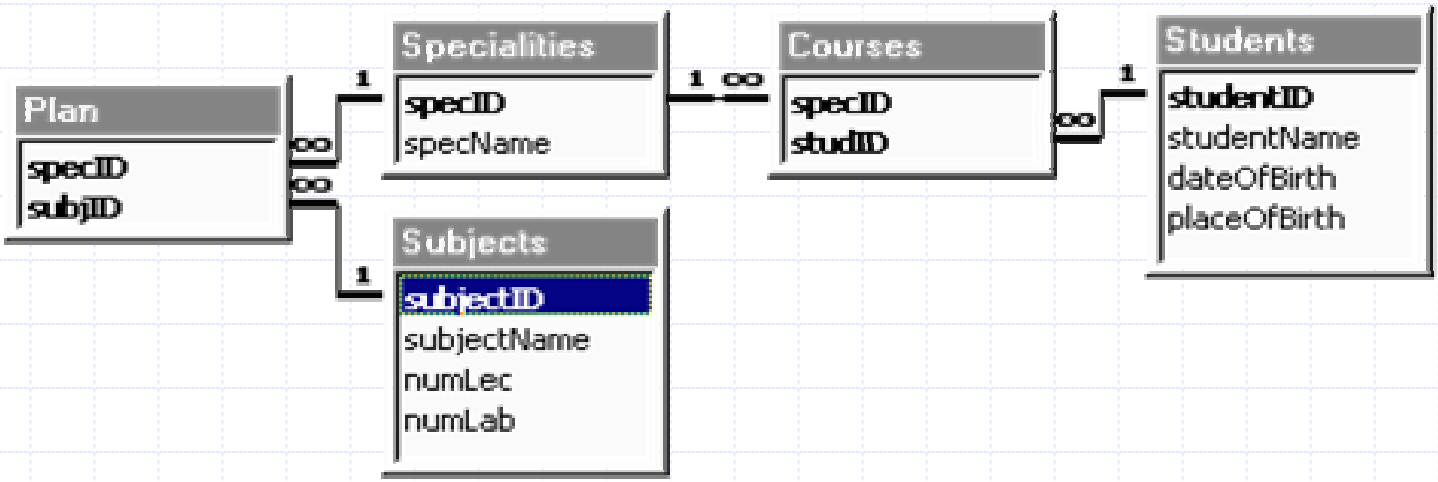
Примери




```
ALTER TABLE Students
    ADD COLUMN dateOfBirth,
    ADD COLUMN placeOfBirth,
    ADD FOREIGN KEY (placeOfBirth) REFERENCES Map;

DROP TABLE Subjects;
```

Примери

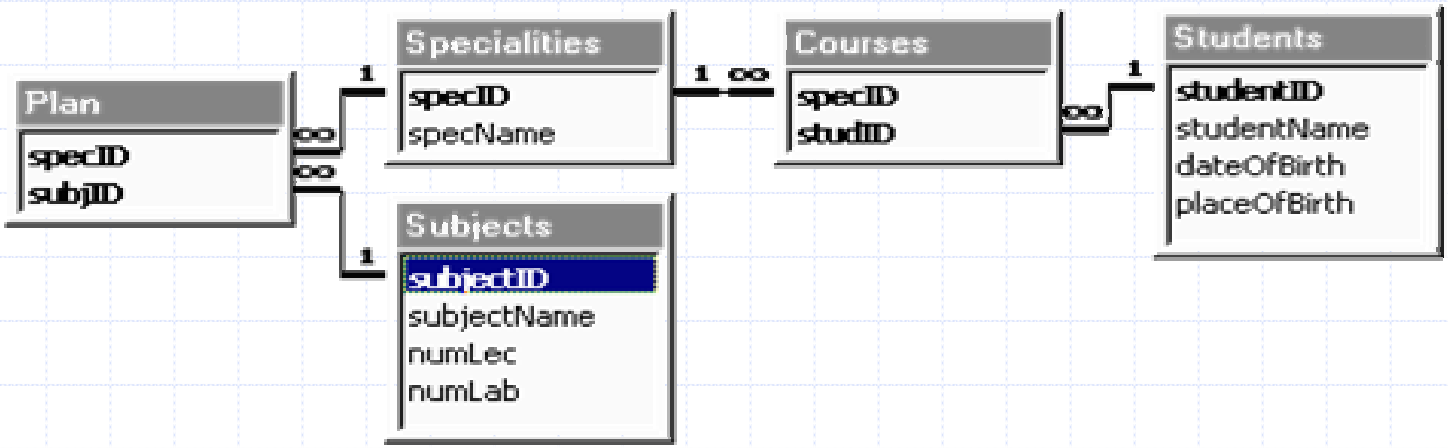


SELECT * FROM Students;

SELECT studentName FROM Students;

SELECT * FROM Students WHERE studentID=14;

Примери

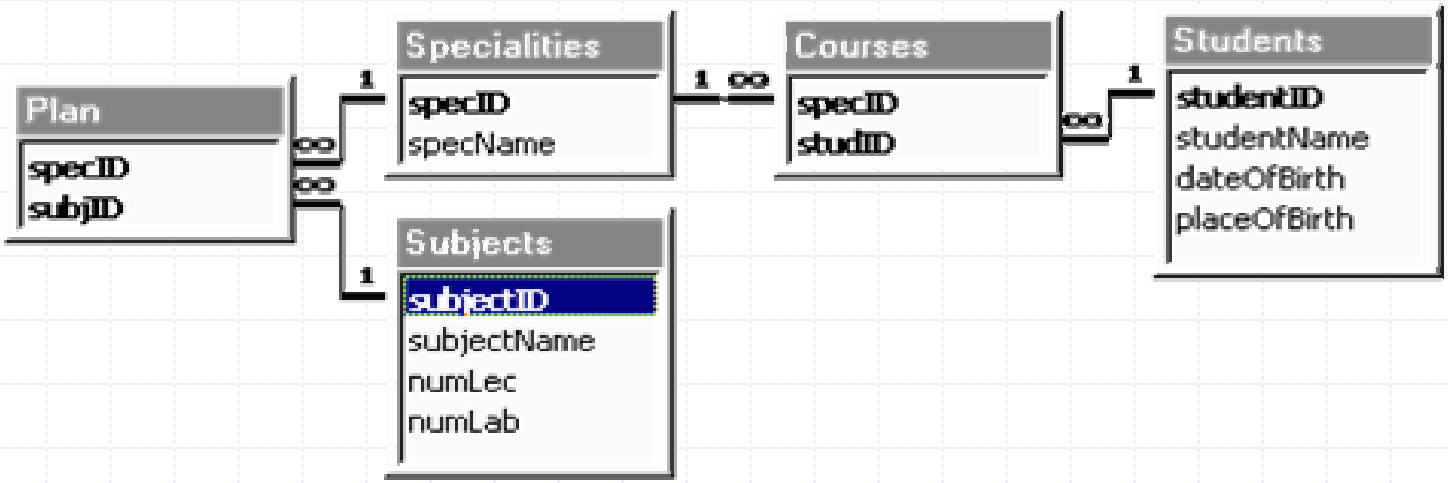



```
SELECT DISTINCT specID FROM Plan;

SELECT studentName FROM Students
        WHERE studentName LIKE "K*";

SELECT * FROM Subjects
        WHERE numLec>20 AND numLab>20;
```

Примери

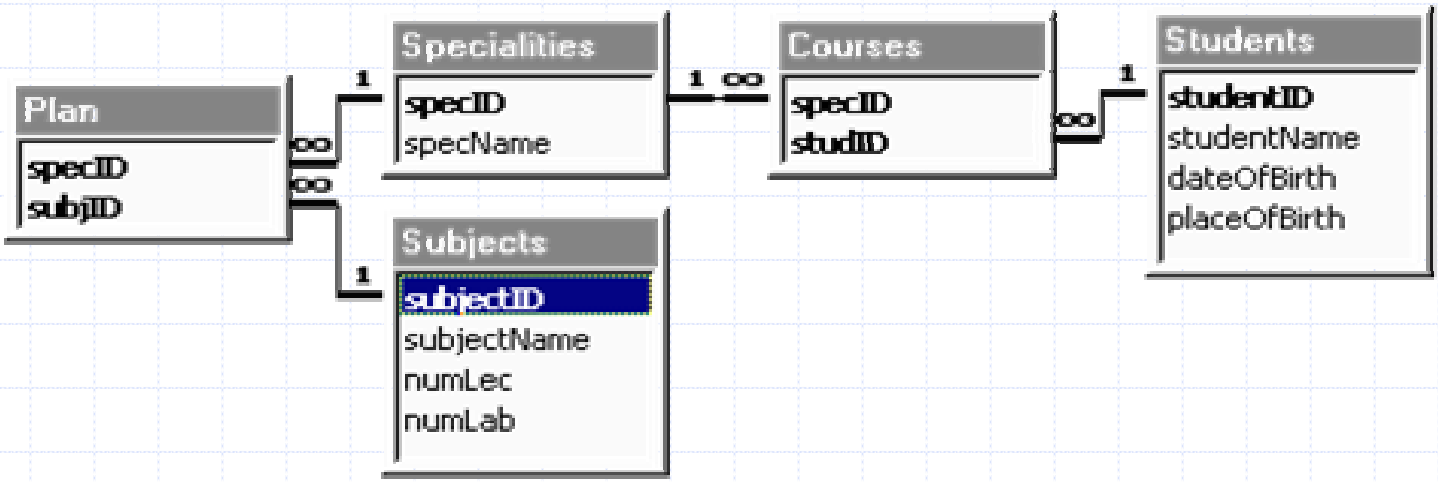



```
SELECT * FROM Students ORDER BY studentName;

SELECT (numLec + numLab)  AS Horarium FROM Subjects;

SELECT * FROM Students JOIN Courses
    ON Students.studentID = Courses.studID;
```

Примери




```
SELECT studentName COUNT (*) FROM Students
        GROUP BY DayPart("yy", dateOfBirth);

SELECT subjectName MIN(numLec + numLab) FROM Subjects
        HAVING numLec >= 10 OR numLab <= 20;

(SELECT * FROM Students )
    UNION / EXCEPT / INTERSECT (SELECT * FROM FootballPlayers);
```

Контрол на данните

<>0

0 Or >100

Like "K???"

<#1/1/96#

>=#1/1/97# And <#1/1/98#

Аритметични операции

= [Orders]![Freight] * 1.1

PrimeFreight: [Freight] * 1.1= [Subtotal] + [Freight]

FreightPercentage: Sum([Freight])/Sum([Subtotal]) *100

=[RequiredDate]-[ShippedDate]

=[EmployeeTotal]/[CountryTotal]

Логически операции

=[Employees]![Country] = "UK"

= [Country] In ("France", "Italy", "Spain") And

Len([PostalCode])<>5

=[Price]*1.06=[Quantity]*[Price]

Операции с дати

=Date()

=Date( )- 30

YearHired: DatePart("yyyy",[HireDate])

=Format(Now(), "ww")

=DatePart("yyyy", [OrderDate])

=DateAdd("y", -10, [PromisedDate])

=DateDiff("d", [OrderDate], [ShippedDate])

Операции с текст

```
="N/A"
FullName: [FirstName] & " " & [LastName]
Address2: [City] & " " & [Region] & " " & [PostalCode]
ProductInitial: Left([ProductName], 1)
TypeCode: Right([AssetCode], 2)
AreaCode: Mid([Phone],2,3)
=Trim([Address])
=IIf(IsNull([Region]), [City] & " "
        & [PostalCode], [City] & " " & [Region] & "  "
        & [PostalCode])
```

Агрегиране и генерализация

=Count(*)

=Count([OrderID])

FreightPercentage: Sum([Freight])/Sum([Subtotal]) *100

=Sum([Sales])

=Sum([Quantity]*[Price])

=[Sales]/Sum([Sales])*100

=Avg([Freight])

Алтернативни стойности

=IIf([Confirmed] = "Yes", "Order Confirmed", "Order Not Confirmed")

=IIf(IsNull([Country]), " ", [Country])

=IIf(IsNull([Region]), [City] & " " & [PostalCode], [City] & " " & [Region] & " " & [PostalCode])

=IIf(IsNull([RequiredDate] - [ShippedDate]), "Check for a missing date", [RequiredDate] - [ShippedDate])

Маски

?	един символ
*	нула или повече символи
#	една цифра
[list]	произволен символ от списъка
[!списък]	произволен символ извън списъка

Използване на маски

ShipName	Like "S*"	започва с ...
ShipName	Like "*Import"	завършва с ...
ShipName	Like "[A-D]*"	започва с някоя от ...
ShipName	Like "*ar*"	съдържа ...
ShipName	Like "Maison Dewe?"	последната буква е неизвестна