QA Light

Python

Михаил

БД и СУБД

Информационная система или база данных

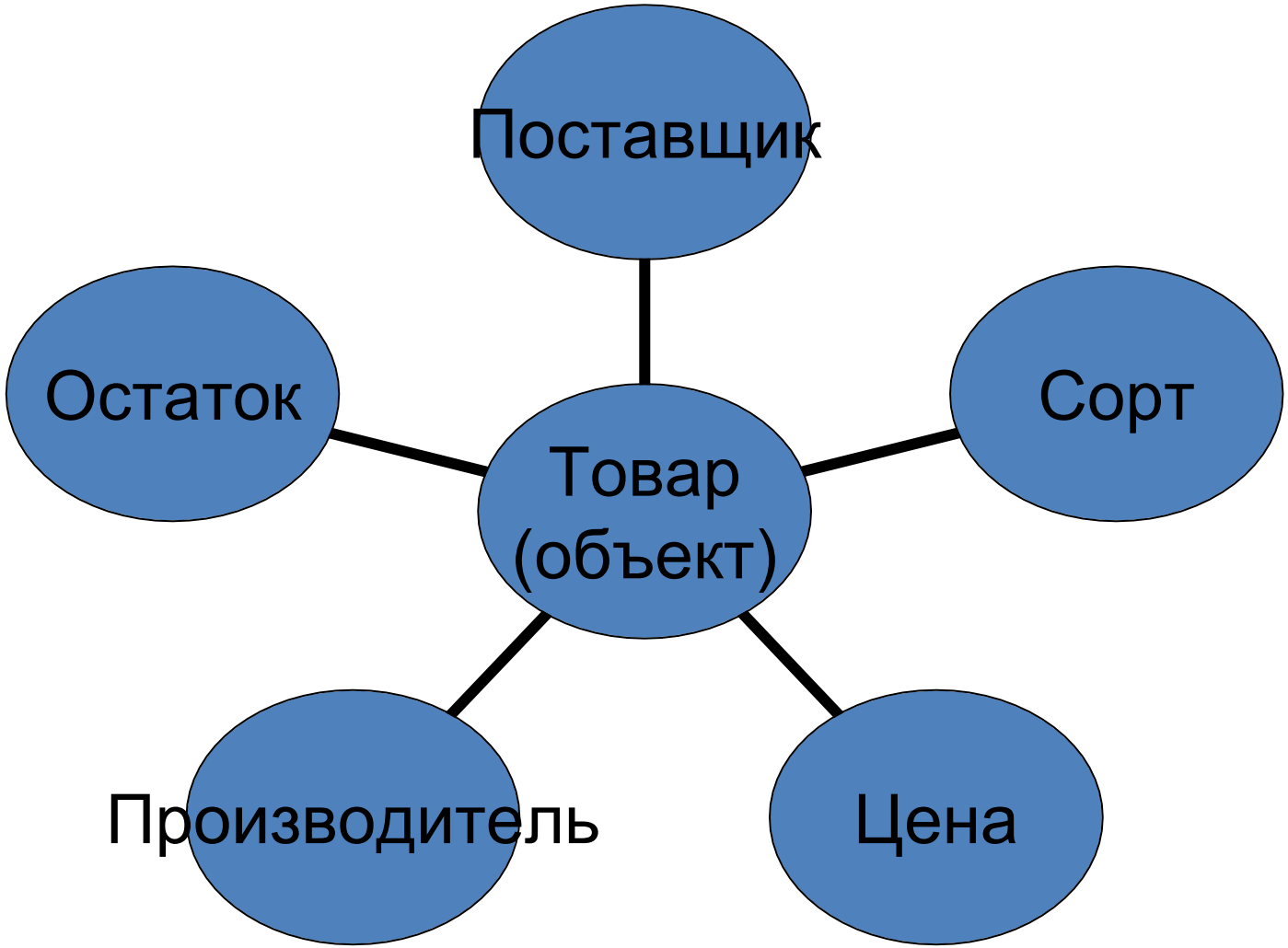
Это большой массив информации (совокупность сведений) о конкретных объектах реального мира в какой-либо предметной области

СУБД (Система управления базами данных

Это совокупность программных средств, обеспечивающая возможность создания базы данных, доступа к данным и управление базой данных.

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

Объект и данное



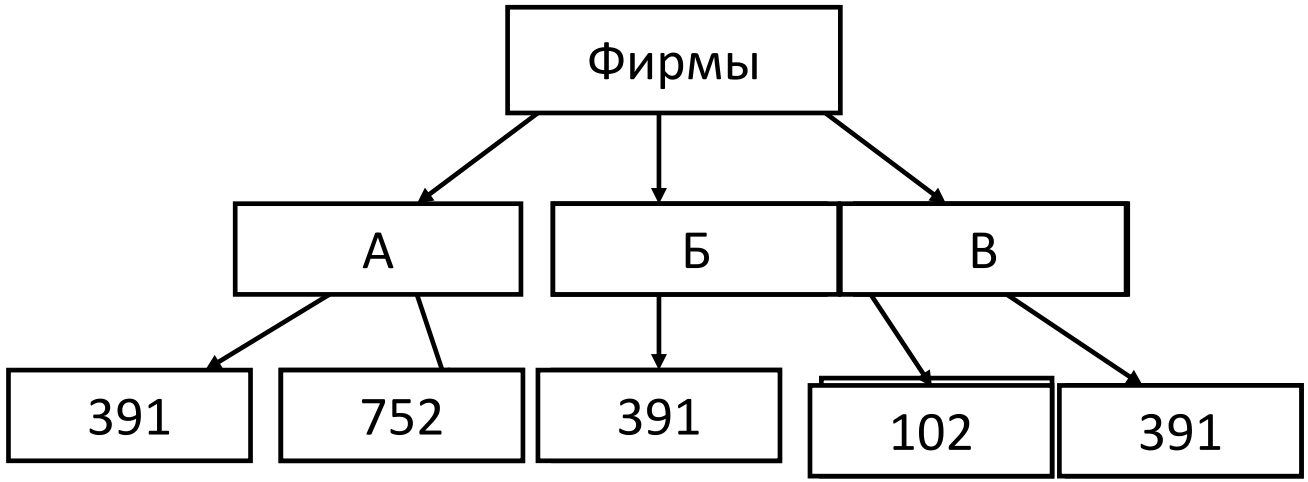
+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

Типы структур БД

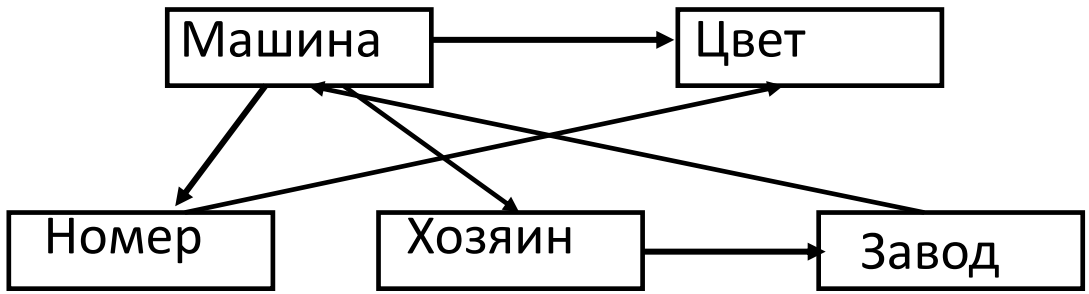
Реляционная

Телефон	ФИО	Адрес
25-25-25	Иванов	Кемерово

Иерархическая



Сетевая



+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

Типы структур БД

Для **иерархических структур** характерна подчиненность объектов нижнего уровня объектам верхнего уровня. В дереве, между верхними и нижними объектами, задано отношение **«один ко многим».** Исходные элементы порождают подчиненные.

Сети имеют много уровней взаимосвязанных объектов, между которыми задано отношение **«многие ко многим».** Сетевая организация обладает большей гибкостью и облегчает процесс поиска требуемых данных.

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

Реляционные базы данных

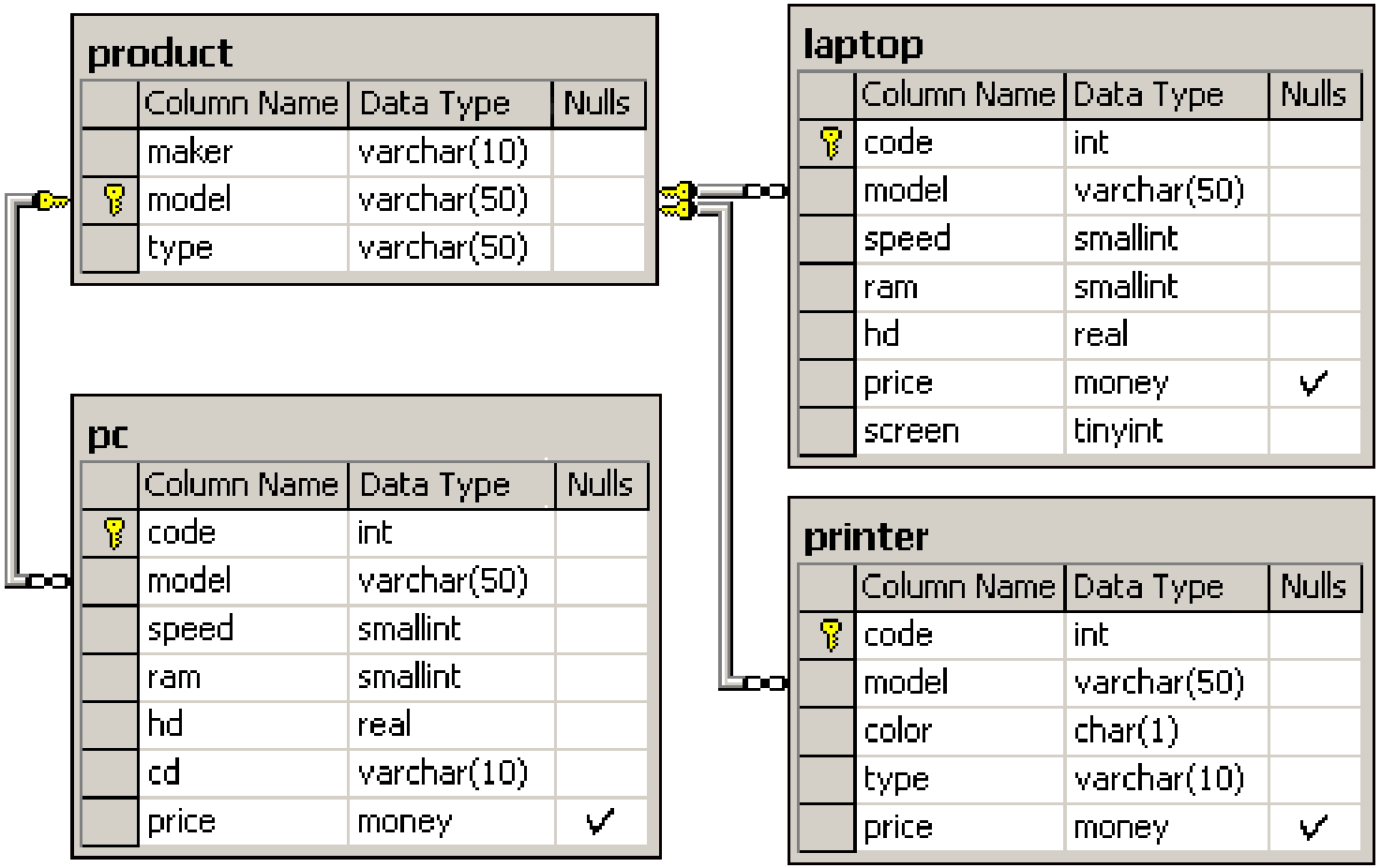
Реляционные базы данных получили наибольшее распространение, т.к. они обладают преимуществом - наглядность и понятность для пользователя табличной структуры.

К реляционной структуре можно свести любой тип структуры данных (деревья и сети).

Название “**реляционная**” (от relational - отношение) связано с тем, что каждая запись в таблице содержит информацию, относящуюся только к одному конкретному объекту.

Чаще всего база данных строится на основе нескольких таблиц, связанных между собой.

www.sql-ex.ru



product

	Column Name	Data Type	Nulls
	maker	varchar(10)	
🔑	model	varchar(50)	
	type	varchar(50)	

laptop

	Column Name	Data Type	Nulls
🔑	code	int	
	model	varchar(50)	
	speed	smallint	
	ram	smallint	
	hd	real	
	price	money	✓
	screen	tinyint	

pc

	Column Name	Data Type	Nulls
🔑	code	int	
	model	varchar(50)	
	speed	smallint	
	ram	smallint	
	hd	real	
	cd	varchar(10)	
	price	money	✓

printer

	Column Name	Data Type	Nulls
🔑	code	int	
	model	varchar(50)	
	color	char(1)	
	type	varchar(10)	
	price	money	✓

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

Типы данных

Примеры названий типов из запросов `CREATE TABLE` или выражения `CAST`	Результирующая аффинированность
INT INTEGER TINYINT SMALLINT MEDIUMINT BIGINT UNSIGNED BIG INT INT2 INT8	INTEGER
CHARACTER(20) VARCHAR(255) VARYING CHARACTER(255) NCHAR(55) NATIVE CHARACTER(70) NVARCHAR(100) TEXT CLOB	TEXT
BLOB *нет указания типа данных*	NONE
REAL DOUBLE DOUBLE PRECISION FLOATNONE	REAL
NUMERIC DECIMAL(10,5) BOOLEAN DATE DATETIME	NUMERIC

CREATE TABLE

Новая таблица создаётся с помощью команды **CREATE TABLE** после которой указывается имя таблицы, а затем в круглых скобках указываются имена столбцов с параметрами.

```
1 CREATE TABLE tbl_info (
2     _id integer PRIMARY KEY AUTOINCREMENT,
3     name text NOT NULL,
4     age integer NOT NULL,
5     city text NOT NULL
6 );
```

Ключевое слово **NOT NULL** требует обязательно присваивать значение столбцу при команде **INSERT INTO**.
Существует также ключевое слово **DEFAULT**, позволяющее вставить значение по умолчанию, например, **city text NOT NULL DEFAULT "Киев"**.

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

DROP TABLE

Удаление таблицы происходит с помощью команды **DROP TABLE**, затем указывается имя таблицы. При желании можно поставить проверочное условие **IF EXISTS**.

```
1 DROP TABLE IF EXISTS tbl_info;
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

INSERT INTO VALUES

Для вставки новой записи в таблицу используется команда **INSERT INTO**, затем указывается имя таблицы, а в скобках имена столбцов. После них идёт ключевое слово **VALUES**, после которого в скобках идут вставляемые значения. Важно соблюдать количество столбцов с вставляемыми данными и их очерёдность при перечислении.

```
1 INSERT INTO tbl_info(name, age, city)
2 VALUES ("Кот Васька", 29, "Киев");
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

SELECT * FROM

Для чтения данных используется команда **SELECT** с условием, затем ключевое слово **FROM** с указанием имени таблицы. Чтобы просмотреть все записи, для условия **SELECT** используется звёздочка (*).

```
1 SELECT * FROM tbl_info;
```

UPDATE

После **UPDATE** указываете таблицу, после **SET** - в каком столбце нужно внести изменения и указывается новое значение, а затем указывается условие.
Можно обновлять группу столбцов, указывая их через запятую.

```
1 UPDATE tbl_info SET age=12 WHERE _id=1;
```

Команда **UPDATE** заменяет собой пару команд **INSERT/DELETE**.
Обновить данные в нужном столбце:

```
1 UPDATE table_name
2 SET имя_столбца = новое_значение
3 WHERE имя_столбца = старое_значение;
```

Также можно производить математические действия: прибавлять, отнимать, умножать, делить. Увеличим возраст кота на день рождения.

```
1 UPDATE tbl_info SET age=age+1 WHERE name="Мурзик";
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

DELETE

Вам не надо перечислять все столбцы, достаточно указать в условии нужный столбец. Условие **WHERE** работает аналогично как в команде **SELECT** и позволяет использовать ключевые слова **LIKE**, **BETWEEN** и т.д.

```
1 DELETE FROM tbl_info WHERE _id=1;
2 DELETE * FROM tbl_info;
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

SELECT

QA Light

Чтобы не искать все записи, можно ограничить поиск условием **WHERE**, после которого идёт имя столбца и условие равенства. Показать всех котов, чей возраст меньше 15.

```
1 SELECT * FROM tbl_info WHERE age < 15;
2 SELECT name FROM tbl_info WHERE age < 15;
3 SELECT name, city FROM tbl_info WHERE age < 15;
4 SELECT name, city FROM tbl_info WHERE age < 15
5 AND city="Киев";
6 SELECT name, city FROM tbl_info WHERE age < 15
7 OR city="Киев";
8 SELECT * FROM tbl_info WHERE age IS NULL;
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

SELECT

Символ **%** в строке указывает на любое слово с нужным окончанием (представляет любое количество неизвестных символов).
Также можно использовать спецсимвол _ для одного символа.
С помощью ключевого слова **BETWEEN** можно быстро и удобно задать диапазон.
С помощью условия **IN** за которыми в скобках идут нужные значения, можно задать нужные параметры. Ключевое слово NOT можно использовать не только с IN, но и с BETWEEN, LIKE.

```
1 SELECT name FROM tbl_info WHERE name LIKE '%ик';
2 select name from tbl_info where name like '_аська';
3 SELECT name FROM tbl_info WHERE age BETWEEN 10 and 20;
4 SELECT name FROM tbl_info WHERE age >= 10 and age <=20;
5 SELECT name FROM tbl_info WHERE age IN (10, 29);
6 SELECT name FROM tbl_info WHERE age = 10 OR age = 29;
7 SELECT name FROM tbl_info WHERE age NOT IN (10);
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

SELECT

Узнать число записей можно через функцию **COUNT**. Если запись содержит **NULL**, то она не учитывается.

Для показа минимального или максимального значения используются функции **MIN** или **MAX.**

Если нам нужно вывести только определённое количество записей, то используйте ключевое слово **LIMIT** с указанием значения.

Существует расширенная версия, когда можно указать два значения через запятую. В первой указывается номер записи (отсчёт от 0), а вторая - число записей. Например, показать вторую запись из таблицы.

```
1 SELECT COUNT(name) FROM table_info;
2 SELECT name, MAX(age) FROM table_info;
3 SELECT * FROM table_info LIMIT 3;
4 SELECT * FROM table_info LIMIT 1,1;
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

ALTER

Добавить новый столбец в таблицу можно с помощью необязательного ключевого слова **ALTER**, за которым идёт название столбца в таблице.
Чтобы указать, после какого столбца нужно добавить новый столбец, используйте ключевое слово **AFTER**.
Другие ключевые
слова: **FIRST**, **BEFORE**, **LAST**, **SECOND**, **THIRD**.

```
1 ALTER TABLE tbl_info ADD COLUMN weight INTEGER;
2 ALTER TABLE tbl_info ADD COLUMN weight INTEGER
3 AFTER age;
4 ALTER TABLE tbl_info RENAME TO table_info;
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

QA Light

Python

Киев
ул. Космонавта Комарова 1
НАУ, корп.11

+38 (097) 78 - 010 - 78
+38 (099) 78 - 010 - 78
+38 (063) 78 - 010 - 78
info@qalight.com.ua