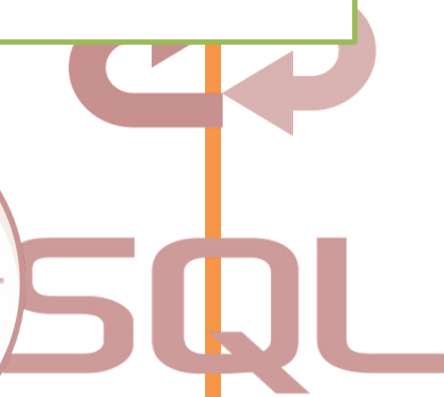


# ПРОГРАММИРОВАНИЕ В РЕЛЯЦИОННОЙ СУБД



SQL

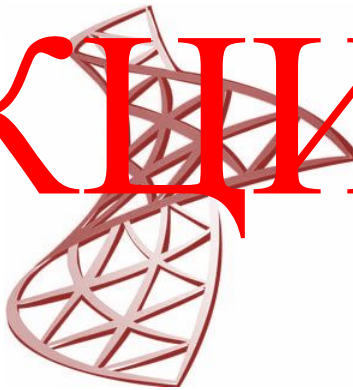
SQL



SQL



# ЛЕКЦІЯ 1



SQL

# ИСТОРИЯ СОЗДАНИЯ SQL



Дональд Чэмбэрлин  
**(Donald D. Chamberlin)**



1974



Рэймонд Бойс  
**(Raymond F. Boyce)**

# СТАНДАРТЫ SQL

Год	Название	Иное название	Изменения
<b>1986</b>	<b>SQL-86</b>	<b>SQL1, SQL-87</b>	Первый вариант стандарта, принятый институтом <b>ANSI</b> и одобренный <b>ISO</b> в <b>1987</b> году
<b>1989</b>	<b>SQL-89</b>	<b>FIPS 127-1</b>	Немного доработанный вариант предыдущего стандарта
<b>1992</b>	<b>SQL-92</b>	<b>SQL2, FIPS 127-2</b>	Значительные изменения ( <b>ISO 9075</b> ); уровень <b>Entry Level</b> стандарта <b>SQL-92</b> был принят как стандарт <b>FIPS 127-2</b>

# СТАНДАРТЫ SQL

Год	Название	Иное название	Изменения
<b>1999</b>	<b>SQL:1999</b>	<b>SQL3</b>	Добавлена поддержка регулярных выражений, рекурсивных запросов, поддержка триггеров, базовые процедурные расширения, не скалярные типы данных и некоторые объектно-ориентированные возможности
<b>2003</b>	<b>SQL:2003</b>		Введены расширения для работы с <b>XML</b> -данными, оконные функции, генераторы последовательностей и основанные на них типы данных



# СТАНДАРТЫ SQL

Год	Название	Иное название	Изменения
<b>2006</b>	<b>SQL:2006</b>		Функциональность работы с <b>XML</b> -данными значительно расширена. Появилась возможность совместно использовать в запросах <b>SQL</b> и <b>XQuery</b> (язык запросов, разработанный для обработки данных в формате <b>XML</b> )
<b>2008</b>	<b>SQL:2008</b>		Улучшены возможности оконных функций, устранены некоторые неоднозначности стандарта <b>SQL:2003</b>

SQL

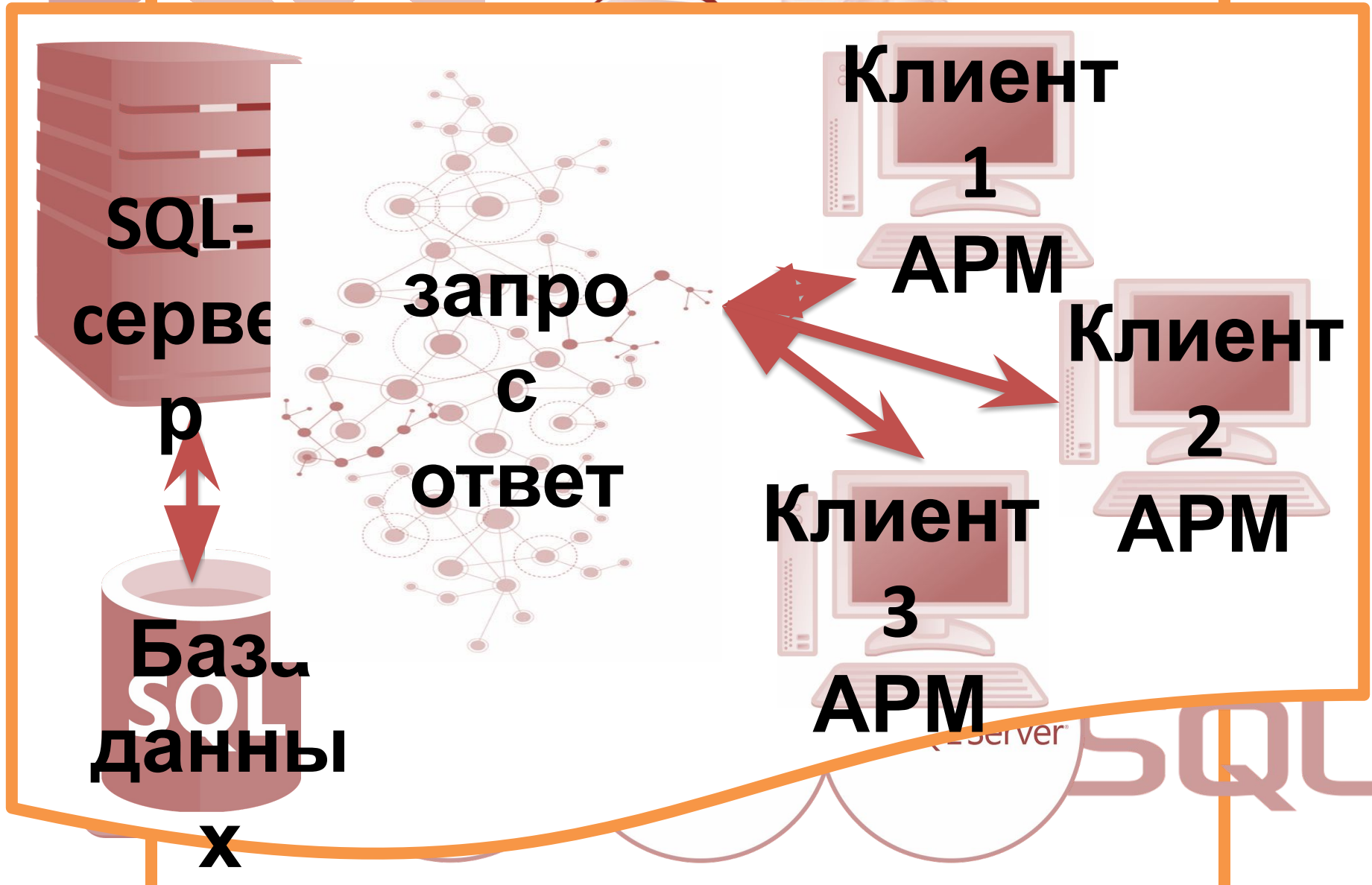
ORACLE®

IBM  
DB2

Microsoft®  
SQL Server®

SQL

# ТЕХНОЛОГИЯ КЛИЕНТ-СЕРВЕР



# ТИПЫ КОМАНД SQL

ОСНОВНЫЕ *КАТЕГОРИИ КОМАНД* ЯЗЫКА **SQL**:

**DDL** – язык определения данных;

**DML** – язык манипулирования данными;

**DQL** – язык *запросов* ;

**DCL** – язык управления данными;

команды администрирования данных;

команды управления транзакциями



# ОПРЕДЕЛЕНИЕ СТРУКТУР БАЗЫ ДАННЫХ (DDL)

Язык определения данных (**Data Definition Language, DDL**) позволяет создавать и изменять структуру объектов *базы данных*, например, создавать и удалять *таблицы*.

Основными командами языка **DDL** являются следующие: **CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, ALTER INDEX, DROP INDEX**

server

SQL

# МАНИПУЛИРОВАНИЕ ДАННЫМИ (DML)

Язык манипулирования данными (***Data Manipulation Language, DML***)

используется для манипулирования информацией внутри объектов *реляционной базы данных* посредством трех основных команд: **INSERT, UPDATE, DELETE**

# ВЫБОРКА ДАННЫХ (DQL)

Язык запросов **DQL** наиболее известен пользователям *реляционной базы данных*, несмотря на то, что он включает всего одну команду **SELECT**. Эта команда вместе со своими многочисленными опциями и предложениями используется для формирования запросов к *реляционной базе данных*

# ЯЗЫК УПРАВЛЕНИЯ ДАННЫМИ (DCL - DATA CONTROL LANGUAGE)

Команды управления данными позволяют управлять доступом к информации, находящейся внутри *базы данных*. Как правило, они используются для создания объектов, связанных с доступом к данным, а также служат для контроля над распределением привилегий между пользователями. Команды управления данными следующие:

**GRANT, REVOKE**

# КОМАНДЫ АДМИНИСТРИРОВАНИЯ ДАННЫХ

С помощью команд администрирования данных пользователь осуществляет контроль за выполняемыми действиями и анализирует операции *базы данных* ; они также полезны при анализе производительности системы. Следует различать **администрирование данных** с **администрированием базы данных**, которое представляет собой общее **управление базой данных** и подразумевает использование команд всех уровней

server

SQL

# КОМАНДЫ УПРАВЛЕНИЯ ТРАНЗАКЦИЯМИ

Существуют следующие команды,  
позволяющие управлять транзакциями *базы  
данных*:

**COMMIT, ROLLBACK, SAVEPOINT,  
SET TRANSACTION**

# ПРЕИМУЩЕСТВА ЯЗЫКА SQL

СТАНДАРТНО  
НЕЗАВИСИМО  
ПРОСТОТА И  
НЕМЕДЛЕННЫЙ ДОСТУП К  
РАЗЛИЧНЫМ ПРЕДСТАВЛЕНИЯМ  
ВОЗМОЖНОСТИ ПРОГРАММНОГО  
ДИНАМИЧЕСКОЕ РАСШИРЕНИЕ И ИЗМЕНЕНИЕ  
СТРУКТУРЫ БД  
ПОДДЕРЖКА АРХИТЕКТУРЫ КЛИЕНТ-  
СЕРВЕР

SQL

# ЗАПИСЬ SQL-ОПЕРАТОРОВ

Все запросы на получение практически любых данных из одной или нескольких таблиц выполняются с помощью единственного предложения **SELECT** .

В синтаксических конструкциях для обращения к БД используются следующие обозначения:

- 1)** звездочка (**\***) для обозначения «все» – употребляется в обычном для программирования смысле, т.е. «все случаи, удовлетворяющие определению»;
- 2)** квадратные скобки (**[]**) – означают, что конструкции, заключенные в эти скобки, являются необязательными (т.е. могут быть опущены);



# ЗАПИСЬ SQL-ОПЕРАТОРОВ

- 3)** фигурные скобки (**{ }**) – означают, что конструкции, заключенные в эти скобки, должны рассматриваться как целые синтаксические единицы, т.е. они позволяют уточнить порядок разбора синтаксических конструкций, заменяя обычные скобки, используемые в синтаксисе **SQL**;
- 4)** многоточие (**...**) – указывает на то, что непосредственно предшествующая ему синтаксическая единица факультативно может повторяться один или более раз;

# ЗАПИСЬ SQL-ОПЕРАТОРОВ

**5)** прямая черта (**|**) – означает наличие выбора из двух или более возможностей. Например, обозначение **ASC|DESC** указывает, можно выбрать один из терминов **ASC** или **DESC**; когда же один из элементов выбора заключен в квадратные скобки, то это означает, что он выбирается по умолчанию (так, **[ASC]|DESC** означает, что отсутствие всей этой конструкции будет восприниматься как выбор **ASC**);

**6)** точка с запятой (**;**) – завершающий элемент предложений **SQL**;

# ЗАПИСЬ **SQL**-ОПЕРАТОРОВ

**7)** запятая (,) – используется для разделения элементов списков;

**8)** пробелы ( ) – могут вводиться для повышения наглядности между любыми синтаксическими конструкциями предложений **SQL**;

**9)** жирные прописные латинские буквы и символы – используются для написания конструкций языка **SQL** и должны (если это специально не оговорено) записываться в точности так, как показано-.....;

# ЗАПИСЬ SQL-ОПЕРАТОРОВ

**10)** строчные буквы используются для написания конструкций, которые должны заменяться конкретными значениями, выбранными пользователем, причем для определенности отдельные слова этих конструкций связываются между собой символом подчеркивания (  );

**11)** термины «таблица» и «столбец» заменяют (с целью сокращения текста синтаксических конструкций) термины «имя\_таблицы», «имя\_столбца», ..., соответственно;

# ЗАПИСЬ SQL-ОПЕРАТОРОВ

**12)** термин «таблица» - используется для обобщения таких видов таблиц, как базовая\_таблица, представление или псевдоним; здесь псевдоним служит для временного (на момент выполнения запроса) переименования и (или) создания рабочей копии базовой\_таблицы (представления). Оператор **SELECT** осуществляет выборку из базы данных и имеет наиболее сложную структуру среди всех операторов языка **SQL**. Простейший оператор **SELECT** выглядит:  
**SELECT \* FROM PC;**

# ЗАПИСЬ SQL-ОПЕРАТОРОВ

Он осуществляет выборку всех записей из объекта БД табличного типа с именем **Рс**. При этом столбцы и строки результирующего набора не упорядочены. Чтобы упорядочить поля результирующего набора, их следует перечислить через запятую в нужном порядке после слова

**SELECT:**

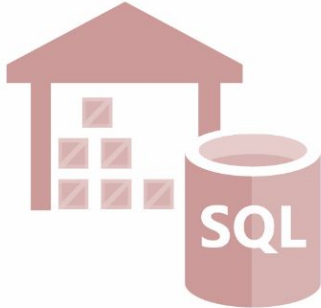
```
SELECT price, speed, hd, ram, cd, model, code  
FROM Pc
```

SQL

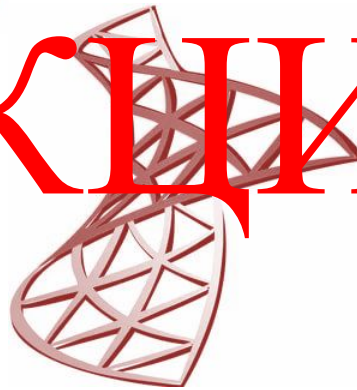
SQL



SQL



# ЛЕКЦІЯ 1



SQL