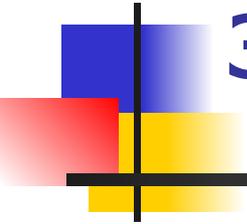
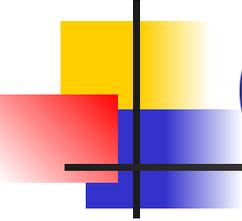


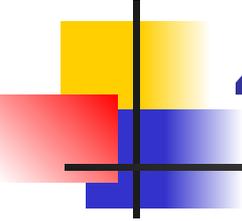
4. Язык структурированных запросов - SQL



4.1 Свойства отношения (таблицы)



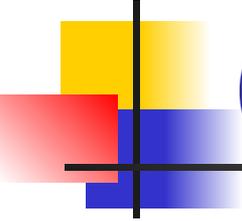
- Идентифицируется уникальным именем;
- Имеет конечное количество столбцов;
- Имеет конечное число строк;
- Столбцы уникальны;
- Столбцы однородны по типу;
- Номер для строки не определен;
- Ячейки могут не содержать значений.



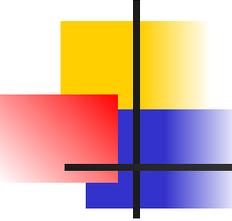
4.2 Операции языка SQL

- Управление таблицами;
- Изменение содержания таблиц;
- Операции с таблицами.

4.3 Управление таблицами (DDL)

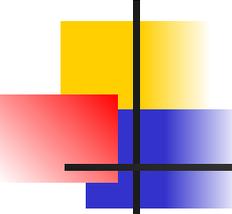


- Создание таблиц;
- Изменение структуры таблиц;
- Удаление таблиц.



4.4 Создание таблиц

```
CREATE TABLE сотрудник(  
    код_сотрудника INT NOT NULL PRIMARY KEY,  
    имя VARCHAR,  
    фамилия VARCHAR NOT NULL;  
    отчество VARCHAR,  
    дата_рождения DATE);
```



4.5 Типы данных

- INT 4
- SMALLINT 2
- FLOAT 4-8
- CHAR(*разм.*)
- VARCHAR (*макс. разм.*)
- BLOB
- DATE
- TIME

CREATE TABLE Совместитель

(код_совместителя INT NOT NULL,

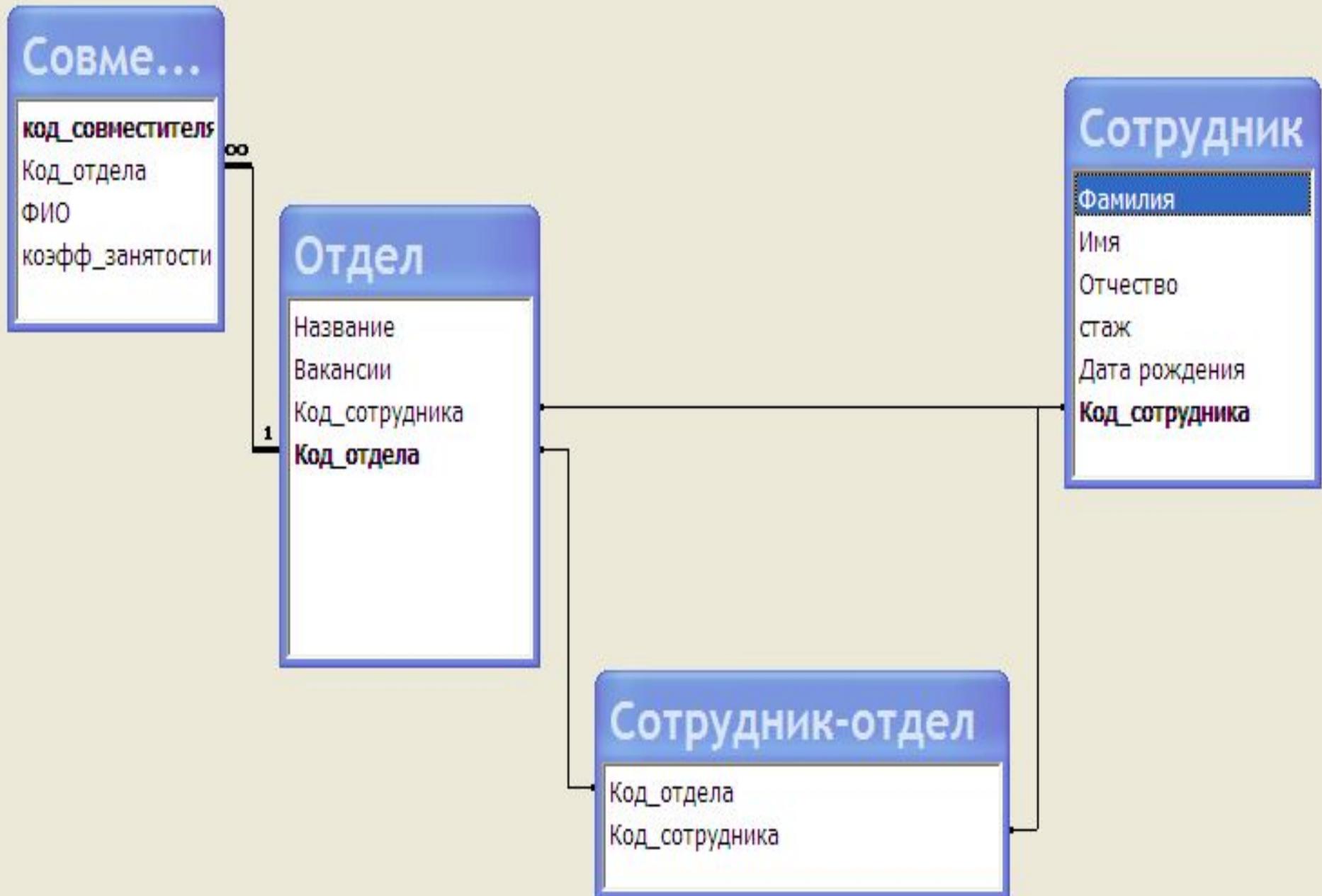
код_отдела INT NOT NULL,

ФИО VARCHAR NOT NULL,

коэфф_занятости INT,

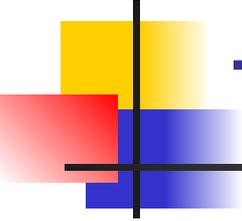
PRIMARY KEY (код_совместителя),

FOREIGN KEY (код_отдела) REFERENCES отдел)



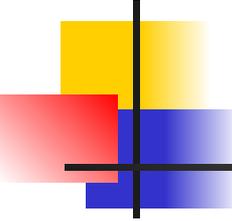
Совместитель : таблица

	Имя поля	Тип данных	
	код_совместителя	Числовой	
	Код_отдела	Числовой	
	ФИО	Текстовый	
	коэфф_занятости	Числовой	



4.6 Изменение структуры таблиц

```
ALTER TABLE сотрудник  
  ADD COLUMN код_отдела INT,  
  ADD COLUMN оклад FLOAT;
```

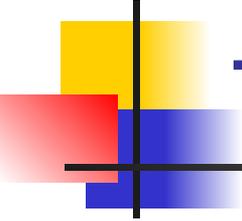


4.7 Удаление таблиц

`DROP TABLE сотрудник;`

`DROP TABLE совместитель RESTRICT;`

`DROP TABLE отдел CASCADE;`



4.8 Изменение содержания таблиц (DML)

INSERT INTO сотрудник

(фамилия, имя, отчество, оклад)

VALUES ('Петров', 'Иван', 'Иванович', 4500);

INSERT INTO сотрудник

(код_совместителя, ФИО, Код_отдела)

SELECT код_совместителя, ФИО, Код_отдела

FROM совместитель

WHERE стаж=5;

UPDATE Совместитель

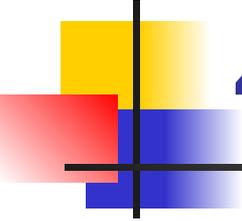
SET разряд=разряд+1

WHERE разряд<7;

DELETE *

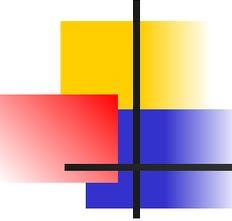
FROM сотрудник

WHERE дата_рождения<01.01.1913;



4.9 Операции с таблицами

- Проекция;
- Ограничение;
- Объединение;
- Декартово произведение.

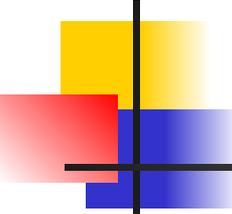


4.10 Проекция

```
SELECT фамилия, имя, отчество FROM сотрудник;
```

```
SELECT DISTINCT имя FROM сотрудник;
```

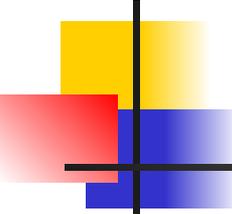
```
SELECT фамилия, имя, отчество  
FROM сотрудник  
ORDER BY фамилия DESC;
```



4.11 Ограничение

```
SELECT *  
FROM сотрудник  
WHERE оклад < 1500;
```

```
SELECT DISTINCT детали.Цвет, детали.Город  
FROM детали  
WHERE детали.Город <> 'Кострома'  
AND детали.Вес > 270  
ORDER BY Город DESC;
```



4.12 Объединение

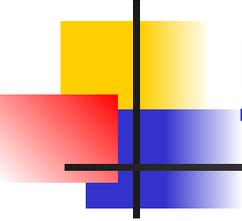
```
SELECT Сотрудник.Фамилия, Сотрудник.Имя,  
Сотрудник.Отчество, Отдел.Название  
FROM Сотрудник INNER JOIN Отдел ON Сотрудник.  
Код_сотрудника = Отдел.Код_сотрудника;
```

Фамилия	Имя	Отчество	Название
Петров	Иван	Иванович	Снабжение
Николаев	Иван	Антонович	Транспорт

4.13 Декартово произведение

SELECT Сотрудник.Фамилия, Сотрудник.Имя,
Сотрудник.Отчество, Праздник.Поздравление
FROM Сотрудник, Праздник;

Фамилия	Имя	Отчество	Поздравление
Петров	Иван	Иванович	С Новым Годом
Сидоров	Анатолий	Петрович	С Новым Годом
Николаев	Иван	Антонович	С Новым Годом
Кошкина	Эльвира	Федоровна	С Новым Годом
Петров	Иван	Иванович	С Днем космонавтики!
Сидоров	Анатолий	Петрович	С Днем космонавтики!
Николаев	Иван	Антонович	С Днем космонавтики!
Кошкина	Эльвира	Федоровна	С Днем космонавтики!
Петров	Иван	Иванович	Мир, Труд, Май!
Сидоров	Анатолий	Петрович	Мир, Труд, Май!
Николаев	Иван	Антонович	Мир, Труд, Май!
Кошкина	Эльвира	Федоровна	Мир, Труд, Май!
Петров	Иван	Иванович	Христос воскрес!
Сидоров	Анатолий	Петрович	Христос воскрес!
Николаев	Иван	Антонович	Христос воскрес!
Кошкина	Эльвира	Федоровна	Христос воскрес!



5. Операции с данными

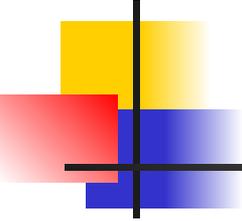
5.1 Арифметические функции

$ABS(x)$ абсолютное значение x

$SQRT(x)$ квадратный корень от x

$MAX(x, y, \dots)$ значение наибольшего
элемента из списка x, y, \dots

$MIN(x, y, \dots)$ значение наименьшего
элемента из списка x, y, \dots



SELECT

[Колич_склад]-[Колич_заказ] AS Expr1,

ABS([Колич_склад]-[Колич_заказ]) AS Expr2

FROM Products

5.2 Строковые функции

 LEFT(s, n) первые n символов строки s

RIGHT(s, n) последние n символов строки s

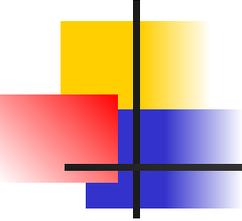
SUBSTRING(s, m, n) строка, получаемая копированием n символов из строки s , начиная с m -ого символа строки s

LCASE(s) строка, полученная из s преобразованием всех букв в строчные

UCASE(s) строка, полученная из s преобразованием всех букв в прописные

CONCAT($s1, s2, \dots$) строка, полученная конкатенацией (слиянием) строк $s1, s2, \dots$

LENGTH(s) длина строки s



```
SELECT
    SUBSTRING([First_Name],1,1] As Initial,
    [Last_Name]
FROM Employees
```

Initial	Last Name
---------	-----------

Н	Васильев
П	Грушко
Д	Утряхина

5.3 Операторы и функции, возвращающие логическое значение (1 – «истина», 0 – «ложь»)

$x = y$ равно

$x \neq y$ не равно

$x > y$ больше

$x < y$ меньше

$x \geq y$ больше или равно

$x \leq y$ меньше или равно

NOT l - 1, если $l=0$, 0, если $l=1$

$l1$ AND $l2$ результат логической операции
"И" над $l1$ и $l2$

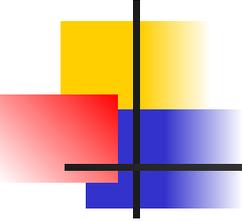
$l1$ OR $l2$ результат логической операции
"ИЛИ" над $l1$ и $l2$

BETWEEN (x, y, z) результат выполнения
логического выражения ($x \geq y$ AND $x \leq z$)

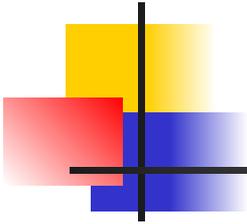
ISNULL (v) 1, если v имеет значение "пусто"
(NULL) 0, в противном случае

s LIKE образец 1, при удачном сопоставлении
строки s с образец или 0, в противном случае

s NOT LIKE образец 0, при удачном
сопоставлении строки s с образец или 1, в
противном случае

- 
-
- `SELECT` Сотрудник.Фамилия, Сотрудник.Имя, Сотрудник.Отчество, Праздник.Поздравление
`FROM` Сотрудник, Праздник
`WHERE` (((Сотрудник.Фамилия) `LIKE` "*на"));

■ 5.4 Агрегативные функции (GROUP BY)



SUM(x) сумма значений столбца x
результатирующей таблицы

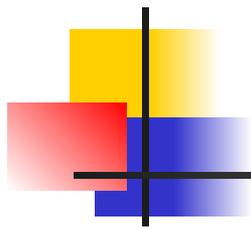
MAX(x) наибольшее значение из всех значений
ячеек столбца x

MIN(x) наименьшее значение из всех значений
ячеек столбца x

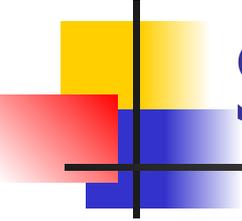
AVG(x) среднее значение для всех значений
ячеек столбца x

COUNT(x) общее количество ячеек в столбце x

- SELECT Код_детали, Наимен,
 - SUM (деталь.Колич) AS Всего
 - FROM деталь
 - GROUP BY Наимен;
-
- SELECT Код_детали, Наимен, Вес, Объем
 - SUM (Вес) AS Полный_вес
 - FROM деталь
 - GROUP BY Наимен;
 - HAVING SUM (Объем) >10;



■ `SELECT Мах(Сотрудник.стаж) FROM Сотрудник;`



SQL3
