

«Базы данных»

Содержание:

1. Основные понятия

2. Отношение

3. Фильтр

4. Логическое выражение

5. Таблицы истинности

6. Запрос

7. Типы соединения таблиц

8. Конструирование базы данных

9. Правила нормализации таблиц

10. Отображения

11. Целостность данных

12. Связь

Что такое Базы данных?

«Множество данных, организованных для быстрого и удобного поиска и извлечения» С. А. Бешенков.

«БД – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области» Ю.А. Шафрин.

«БД – автоматическое справочное бюро, ориентированное на широкий круг заранее не регламентированных запросов и большие массивы однородной информации» А.Г. Гейн.

Отношение

- Андрей – отец Павла
- Сергей – отец Жени
- Андрей – отец Ани
- Папа купил книгу
- Папа купил машину
- Мама купила шубу

Объект X – отец объекта Y

Объект X купил объект Y

Имя отношения

Аргумент

БЫТЬ_ОТЦОМ(X,Y)

КУПИТЬ (X,Y)

БЫТЬ_ОТЦОМ(отец:X,ребенок:Y)

Имя аргумента - атрибут

Отношение - таблица

Отношение - функция

БЫТЬ_ОТЦОМ

Отец	Ребенок
Андрей	Павел
Андрей	Аня
Сергей	Женя

$f(x,y)$

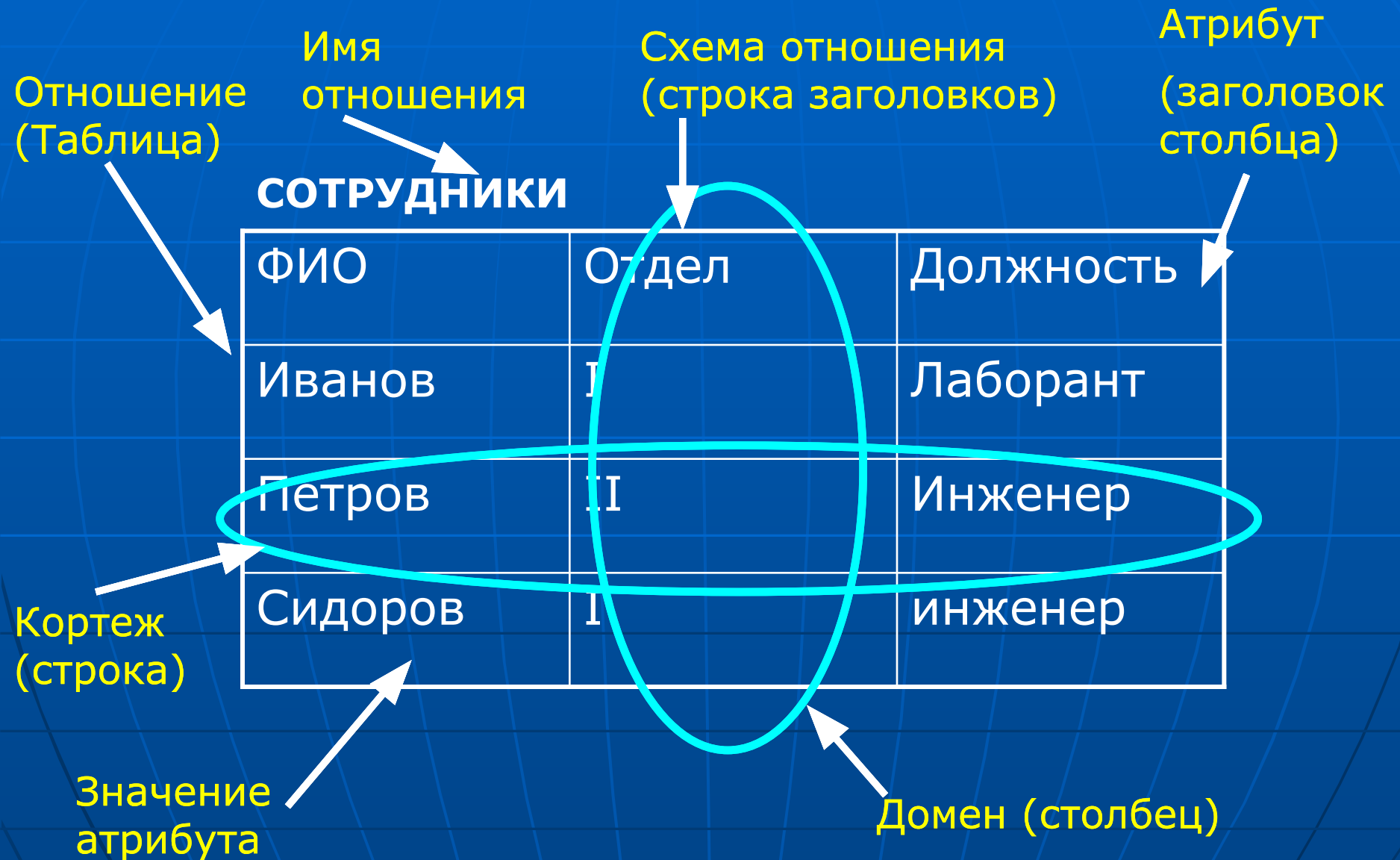
Если каждому значению одной переменной величины X каким-либо образом сопоставлено не более одного значения другой величины Y , то говорят, что задана функция.

КУПИТЬ

Кто	Что
Папа	Книга
Папа	Машина
Мама	Шуба

Таблица – это отношение значений различных атрибутов, относящихся к одному объекту.

Отношение



Что такое СУБД?

СУБД – это оболочка, позволяющая изменять, сохранять информацию в БД, проводить выборку нужной информации по запросу пользователя.

Фильтр

«Совокупность условий (логическое выражение), которую указывает пользователь для выделения некоторого подмножества строк таблицы в реляционной базе данных» Ю.А.Шафрин.

«Фильтр – это набор условий»,

«...режим, позволяющий отображать только некоторые записи»,

«нечто, отсеивающее определенные элементы».

Фильтр – логическое выражение Фильтрация – отбор строк таблицы, для которых данное логическое выражение истинно. А.Г.Гейн

БЫТЬ_ОТЦОМ

Отец	Ребенок
Андрей	Павел
Андрей	Аня
Сергей	Женя
Андрей	Володя

Результат фильтрации:

БЫТЬ_ОТЦОМ

Отец	Ребенок
Андрей	Павел
Андрей	Аня
Андрей	Володя

Фильтр

Исходное отношение:
БЫТЬ_ОТЦОМ(отец, ребенок).

Фильтр:

X(Отец=Андрей)

Результат фильтрации -
отношение:
БЫТЬ_ОТЦОМ(Павел, Женя, Володя).

Фильтр

Фильтром для данного отношения является логическое выражение, не содержащее атрибутов, не входящих в данное отношение.

Результатом фильтрации данного отношения относительно заданного логического выражения (функции X) называется отношение, состоящее только из тех записей исходного отношения, для которых функция X принимает значение ИСТИНА.

А.Г.Гейн

Логическое выражение

«Выражение. Форма записи некоторого высказывания. Выражение состоит из операндов, соединенных специальными знаками, которые определяют семантику выражения» Ю.А.Шафрин.

«Любая комбинация операторов, констант, функций и др. , результатом которой является конкретное значение»
Help Access.

Арифметическое выражение

Формально арифметическое выражение определяется по следующим признакам:

1. Всякое число или атрибут есть арифметическое выражение.
2. Если A арифметическое выражение, то (A) тоже арифметическое выражение.
3. Если A и B арифметические выражения, то $A+B$, $A-B$, $A*B$, A/B тоже арифметические выражения.
4. Других арифметических выражений нет.

Значением арифметического выражения может быть число или арифметическое выражение.

Логическое выражение

Пусть X – атрибут, T – арифметическое выражение.

- Выражение $X=T$, $X<T$, $X>T$, $X<>T$ – логические выражения.
- Если X логическое выражение, то (X) – тоже логическое выражение.
- Если X и Y – логические выражения, то X AND Y , X OR Y , NOT X – тоже логические выражения.
- ИСТИНА и ЛОЖЬ – логические выражения.
- Других логических выражений нет.
- **Значением логического выражения является ИСТИНА и ЛОЖЬ.**

Таблицы истинности логических выражений двузначной логики

AND

X	Y	X and Y
Истина	Истина	Истина
Истина	Ложь	Ложь
Ложь	Истина	Ложь
Ложь	Ложь	Ложь

OR

X	Y	X OR Y
Истина	Истина	Истина
Истина	Ложь	Истина
Ложь	Истина	Истина
Ложь	Ложь	Ложь

NOT

X	NOT X
Истина	Ложь
Ложь	Истина

Таблица истинности сложных логических выражений двузначной логики

X	Y	X OR Y	NOT(X OR Y)	NOT X	NOT X OR Y
Истина	Истина	Истина	Ложь	Ложь	Истина
Истина	Ложь	Истина	Ложь	Ложь	Ложь
Ложь	Истина	Истина	Ложь	Истина	Истина
Ложь	Ложь	Ложь	Истина	Истина	Истина

Порядок выполнения логических операций

Сначала выполняются операции в скобках, затем операция NOT, затем AND и последней OR.

Таблицы истинности логических выражений для трехзначной логики

AND

X	Y	X AND Y
Истина	Истина	Истина
Истина	Не опред	Не опред
Истина	Ложь	Ложь
Не опред	Истина	Не опред
Не опред	Не опред	Не опред
Не опред	Ложь	Ложь
Ложь	Истина	Ложь
Ложь	Не опред	Ложь
Ложь	Ложь	Ложь

X	NOT X
Истина	Ложь
Не определ	Не определ
Ложь	Истина

OR

X	Y	X OR Y
Истина	Истина	Истина
Истина	Не опред	Истина
Истина	Ложь	Истина
Не опред	Истина	Истина
Не опред	Не опред	Не опред
Не опред	Ложь	Не опред
Ложь	Истина	Истина
Ложь	Не опред	Не опред
Ложь	Ложь	Ложь

Таблицы истинности сложных логических выражений для трехзначной логики

X	Y	X OR Y	NOT(X OR Y)	NOT X	NOT X OR Y
Истина	Истина	Истина	Ложь	Ложь	Истина
Истина	Не опр	Истина	Ложь	Ложь	Не опр
Истина	Ложь	Истина	Ложь	Ложь	Ложь
Не опр	Истина	Истина	Ложь	Не опр	Истина
Не опр	Не опр	Не опр	Не опр	Не опр	Не опр
Не опр	Ложь	Не опр	Не опр	Не опр	Не опр
Ложь	Истина	Истина	Ложь	Истина	Истина
Ложь	Не опр	Не опр	Не опр	Истина	Истина
Ложь	Ложь	Ложь	Истина	Истина	Истина

Темы исследовательских заданий учащимся по теме БД

1. Построить таблицы истинности двузначной и трехзначной логики для другого сочетаний простых логических функций.
2. Как выбрать те строки, для которых значение функции X равно **ЛОЖЬ**?
3. Можно ли в трехзначной логике с помощью логических операций построить функцию, которая бы значение **НЕ ОПРЕДЕЛЕНО** переработало в значение **ИСТИНА**.

Функция **NULL** в БД Access

Таблица истинности

X	NULL X
Истина	Ложь
Не определено	Истина
Ложь	Ложь

Бланк QBE (Query By Example-запрос по образцу)

РазговорыФильтр1 : фильтр

Разговоры

- *
- Код**
- Номер телеф
- Дата
- Город

Поле:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Сортировка:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Условие отбора:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
или:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Дизъюнктивная нормальная форма логических выражений

**(Вес=2 AND NOT (Высота>3) AND (Ширина<4))
OR (Вес=3 AND Высота>5 AND NOT (Ширина<7))
OR (Вес>4 AND Высота>6 AND Ширина<8).**

Поле	Вес	Высота	Ширина
Условие отбора	=2	NOT>3	<4
Или	=3	>5	NOT<7
Или	=4	>6	<8

Дизъюнктивная нормальная форма логических выражений

(Возраст >80 AND NOT (Зарплата <100))
OR (NOT (Возраст >80) AND Зарплата <100)

Поле	Возраст	Зарплата
Условие отбора	>80	NOT <100
Или	NOT >80	<100

Не ДНФ выражения

NOT((Возраст >80 AND NOT (Зарплата <100))

OR (NOT (Возраст >80) AND Зарплата <100))

Поле	Возраст	Зарплата
Условие отбора		
Или		

(Возраст >80 OR NOT (Зарплата <100))

AND (NOT (Возраст >80) AND Зарплата <100)

Алгоритм приведения логических выражений к ДНФ виду

1. Записать таблицу истинности, перебирая возможные варианты значений
2. Для каждой строки, где $F(x)$ ИСТИНА, записать AND выражение
3. Все AND выражения соединить оператором OR.

Приведение логического выражения к ДНФ виду

(Возраст >80 OR NOT (Зарплата <100)) AND (NOT (Возраст >80) AND Зарплата <100)

Примем Возраст >80 – ИСТИНА, Зарплата <100 – ИСТИНА

1	2	3	4	5	6	7
Возраст	Зарплата	NOT (2)	NOT(1)	OR(1,3)	AND 4,2	AND 5,6
Истина	Истина	Ложь	Ложь	Истина	Ложь	Ложь
Ложь	Истина	Ложь	Истина	Ложь	Истина	Ложь
Истина	Ложь	Истина	Ложь	Истина	Ложь	Ложь
Ложь	Ложь	Истина	Истина	Истина	Ложь	Ложь

Пример 1. **(Возраст >80 OR NOT (Зарплата <100)) AND (NOT (Возраст >80) AND Зарплата <100)**

Исходная таблица

Фамилия	Возраст	Зарплата
Иванов	90	<100
Петров	70	<100
Сидоров	90	>100
Кузьмин	70	>100

1 - OR

Фамилия	Возраст	Зарплата
Иванов	90	<100
Сидоров	90	>100
Кузьмин	70	>100

2 - AND

Фамилия	Возраст	Зарплата
Петров	70	<100

3 - AND (1,2)

Записей, удовлетворяющих такому фильтру не может существовать

Пример 2

NOT((Возраст >80 AND NOT (Зарплата <100)) OR (NOT (Возраст >80) AND Зарплата <100))

Примем: Возраст >80 – ИСТИНА, Зарплата <100 – ИСТИНА;

1	2	3	4	5	6	7	8
Возраст	Зарплата	NOT 2	NOT 1	AND 1,3	AND 4,2	NOT 5	OR 7,6
Истина	Истина	Ложь	Ложь	Ложь	Ложь	Истина	Истина
Ложь	Истина	Ложь	Истина	Ложь	Истина	Истина	Истина
Истина	Ложь	Истина	Ложь	Истина	Ложь	Ложь	Ложь
Ложь	Ложь	Истина	Истина	Ложь	Ложь	Истина	Истина

Пример 2 Примем: $X := \text{Возраст} > 80$ – ИСТИНА,
 $Y := \text{Зарплата} < 100$ – ИСТИНА;

1	2	8
Воз- раст	Зар- плата	OR 7,6
Истина	Истина	Истина
Ложь	Истина	Истина
Истина	Ложь	Ложь
Ложь	Ложь	Истина

**(X AND Y)
OR (NOT X AND Y)
OR (NOT X AND NOT Y);**

(Возраст >80 AND Зарплата <100)

OR (NOT(Возраст >80) AND Зарплата <100)

**OR (NOT(Возраст >80) AND NOT (Зарплата
<100)).**

Пример 2
проверка

(Возраст >80 AND Зарплата <100)
OR (NOT(Возраст >80) AND Зарплата <100)
OR (NOT(Возраст >80) AND NOT (Зарплата <100))

Исходная таблица

Фамилия	Возраст	Зарплата
Иванов	90	<100
Петров	70	<100
Сидоров	90	>100
Кузьмин	70	>100

1 - AND

Фамилия	Возраст	Зарплата
Иванов	90	<100

2 - AND

Фамилия	Возраст	Зарплата
Петров	70	<100

OR (1,2,3)

Фамилия	Возраст	Зарплата
Иванов	90	<100
Петров	70	<100
Кузьмин	70	>100

3 - AND

Фамилия	Возраст	Зарплата
Кузьмин	70	>100

Пример 2 NOT((Возраст >80 AND NOT (Зарплата <100)) OR (NOT (Возраст >80) AND Зарплата <100))

проверка

Исходная таблица

Фамилия	Возраст	Зарплата
Иванов	90	<100
Петров	70	<100
Сидоров	90	>100
Кузьмин	70	>100

4- OR(2,3)

Фамилия	Возраст	Зарплата
Иванов	90	<100
Петров	70	<100
Кузьмин	70	>100

1 - AND

Фамилия	Возраст	Зарплата
Сидоров	90	>100

2 - NOT(1)

Фамилия	Возраст	Зарплата
Иванов	90	<100
Петров	70	<100
Кузьмин	70	>100

3 - AND

Фамилия	Возраст	Зарплата
Петров	70	<100

Вывод

Каждое логическое выражение
может быть преобразовано к
равносильному
дизъюнкционному нормальному
выражению.

NULL значение

Работнички

Фамилия	Имя	Отчество
Иванов	Иван	Петрович
Петров	Иван	Семенович
Сидоров	Петр	Иванович
Кузьмин	Иван	

Имя=Иван AND NOT Отчество=Петрович

Имя=Иван AND Отчество=NULL

Что такое Запрос?

Логическое выражение, задающее условие для отбора данных:
формула запроса.

Объект СУБД, таблица-запрос, в которой хранятся данные, удовлетворяющие формуле запроса.

В БД Access первое соответствует структурному виду запроса, второе – табличному.

Определения понятия «Запрос»

Запрос – это вопрос о данных. ...запрос –выборка – это производная таблица». Ю. А. Шафрин

«Запросы – это специальные структуры, предназначенные для обработки данных.....Запрос – это специальный объект БД, предназначенный для выборки данных из таблицы» С. В. Симонович

“Запрос – это требование на отбор данных или на выполнение определенного действия с данными». Help Access

«Запрос есть функция, сопоставляющая ответ каждому допустимому состоянию»

А. Г. Гейн

Последовательность операций при составлении ответа на запрос

1. Соединение таблиц
2. Фильтрация данных
3. Удаление ненужных столбцов – проекция таблицы.

Типы соединений таблиц:

внутреннее соединение

ТЕЛЕФОНЫ

ЗВОНКИ

Номер	Владелец	Адрес
411064	Борисов	Чекистов 1
473298	Алексеев	Сиреневый 5
56677	Петров	Ленина 17
601130	Иванов	Щорса 50

Кто	Кому	Дата
411064	473298	08.08.2003
473298	601130	08.08.2003
550011	330102	08.08.2003

Соединяются строки с одинаковыми значениями атрибутов

Внутреннее соединение по условию **НОМЕР=КТО**

Номер	Владелец	Адрес	Кто	Кому	Дата
411064	Борисов	Чекистов 1	411064	473298	08.08.2003
473298	Алексеев	Сиреневый 5	473298	601130	08.08.2003

Типы соединений таблиц:

внешнее соединение

Внешнее левое

Номер	Владелец	Адрес	Кто	Кому	Дата
411064	Борисов	Чекистов 1	411064	473298	08.08.2003
473298	Алексеев	Сиреневый 5	473298	601130	08.08.2003
56677	Петров	Ленина 17			
601130	Иванов	Щорса 50			

Внешнее правое

Номер	Владелец	Адрес	Кто	Кому	Дата
411064	Борисов	Чекистов 1	411064	473298	08.08.2003
473298	Алексеев	Сиреневый 5	473298	601130	08.08.2003
			550011	330102	08.08.2003

Соединение по разными атрибутам

ТЕЛЕФОНЫ

Номер	Владелец	Адрес
411064	Борисов	Чекистов 1
473298	Алексеев	Сиреневый 5
56677	Петров	Ленина 17
601130	Иванов	Щорса50

ЗВОНКИ_1

Кто	Кому	Дата
551718	557181	10.10.2002

Таблицы можно соединять по условиям
НОМЕР-КТО и
НОМЕР-КОМУ

Закономерности обработки таблиц

1. Пусть A и B – отношения. Тогда $A \otimes B = B \otimes A$, где \otimes означает соединение.
2. Пусть A, B, C – отношения. Тогда $A \otimes (B \otimes C) = (A \otimes B) \otimes C$
3. Пусть A – отношение, Φ_1, Φ_2 - фильтры.
 $A | \Phi_1 | \Phi_2 = A | \Phi_2 | \Phi_1$, т.е. очередность фильтров можно переставлять.
4. Пусть A – отношение, $A/$ - проекция.
 $A/ | \Phi = \Phi | A/$, т.е. очередность фильтра и проекции можно переставлять.
5. Пусть A – отношение, a_1, a_2 – атрибуты. $A/ a_1, a_2 = A/ a_2, a_1$, т.е. при выполнении проекции не имеет значения порядок атрибутов.

Конструирование базы данных

Данные и информация

Данные – это статистические значения, хранящиеся в таблицах БД.

Информация – это сведения, которые запрашиваются пользователем и предоставляются ему в наиболее удобном виде.

Данные хранятся

Информация запрашивается

Прежде, чем разрабатывать структуру БД, необходимо определить задачи (какую информацию надо будет получать из БД), а затем продумать, что нужно хранить в БД (какие данные нужны для получения требуемой информации)

Рабочий бланк «Задачи»

Название задачи:

Краткое описание:

Связанные задачи:

Элемент данных	Использование	Описание	Объект

Рабочий бланк «Задачи» БД Сессия

Название задачи: Создать ведомость получения стипендии

Краткое описание: Отбор студентов на повышенную стипендию (200%), имеющих все «5», на стипендию 150% за 1 «4» и остальные «5» и на стипендию 100% за две и более «4»

Связанные задачи: : 1. Ввод данных о студентах; 2. Ввод оценок за экзамены; 3. Анализ оценок. 4. Создание отчета с ведомостью

Элемент данных	Использование	Описание	Объект
Номер зачетки	I	Номер зачетки	Табл. Студент
Фамилия	I	Фамилия	Табл. Студент
Номер зачетки	I	Номер зачетки	Табл. Студент
Номер зачетки	O	Номер зачетки	Табл. Экзамен
Фамилия	O	Фамилия	Запросы

Варианты использования данных:

I – Input – данные вводятся в таблицу

O- Output – данные считываются из другой таблицы

U – Update – данные изменяются в процессе обработки

D – Delete – данные удаляются в процессе работы

C – Calculate – данные вычисляются

Диаграмма взаимосвязей задач.

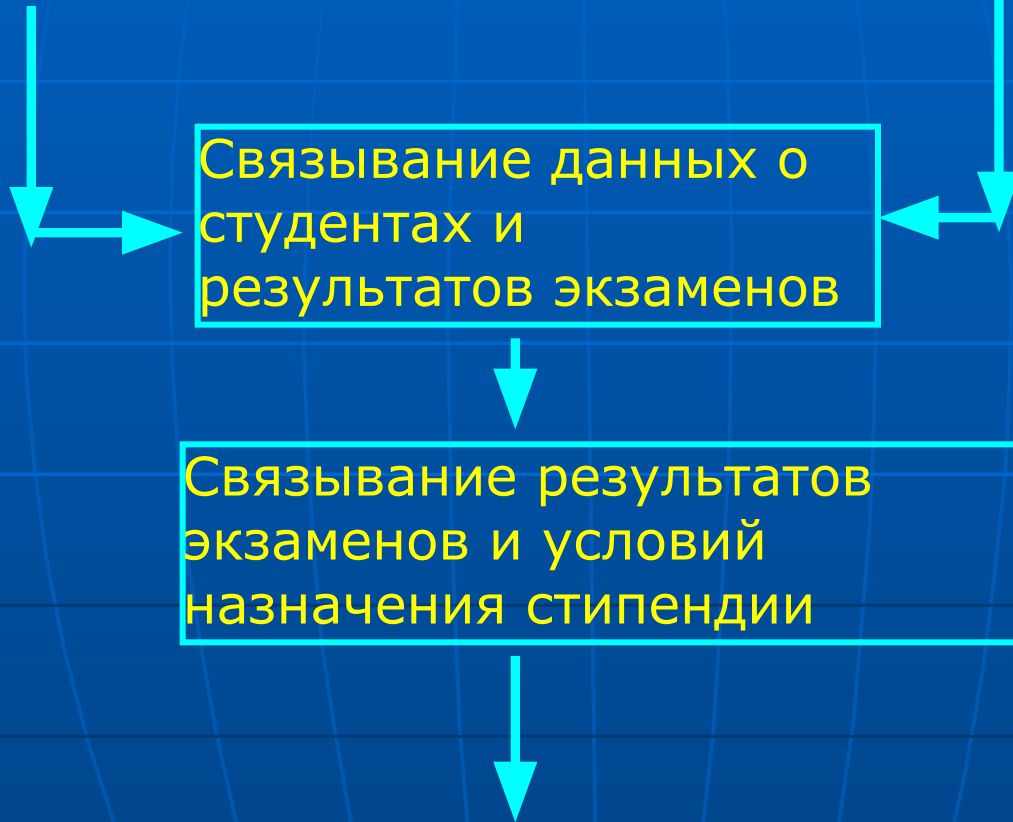
Ввод данных о студентах

Ввод результатов экзаменов

Связывание данных о студентах и результатов экзаменов

Связывание результатов экзаменов и условий назначения стипендии

Создание отчета о стипендии



Рабочий бланк «Объекты»

Имя объекта: Таблица Студент

Краткое описание: Информация о студентах

Связанные объекты: Имя
Табл. Экзамены
Тип связи
Один

Элемент данных	Тип данных	Описание	Условие на значение
Номер зачетки	Числовой (Целое)	Номер зачетки	Обязательное (ключ)
Фамилия	Текстовый	Фамилия (20)	Обязательное

Описание данных

Поле	Тип данных	Размер поля	Формат поля	Число десятичных знаков	Обязательное поле	Пустые строки	Индексированное поле
Номер зачетки	Числовой	Целое	Основной	0	Да	нет	Да (Совпадение не доп.)
Фамилия	Текстовый	25			Да	Нет	Да (Совпад. допуск)

NULL значение и пустые строки

Пустые строки Да	Пустые строки Нет
Значение поля известно, но поле пустое	Значение поля не известно, NULL
Пустые строки равны друг другу	NULL строки не равны друг другу
Таблицы можно соединять по пустым строкам	Таблицы нельзя соединять по NULL строкам
	Значение NULL не учитывается в итоговых функциях
Ввод пустой строки: знак «» или пробел	Незаполненные строки автоматически преобразуются в NULL строки

Первичный ключ

«Поле, которое однозначно определяет соответствующую запись» Макарова.

«...идентификатор, значение которого однозначно определяет эту и только эту запись» Ю.А. Шафрин

«...одно или несколько полей, комбинация значений которого однозначно определяет каждую запись в таблице. » help Access.

«...ключ – это один из возможных ключей, назначаемых пользователем. Возможным ключом называется набор атрибутов данного отношения, если отображение, порождаемое этим набором из данного отношения является ключом» А.Г.Гейн.

Правила нормализации таблиц

Правило 1. Уникальность полей.

Каждое поле таблицы должно представлять уникальный тип информации, т.е. необходимо избавляться от повторяющихся атрибутов и разделить составные атрибуты на отдельные элементы.

Правило 2. Первичные ключи

Каждая таблица должна иметь уникальный идентификатор или первичный ключ, который может состоять из одного или нескольких полей. Если возможно, в качестве первичного ключа использовать самые простые данные, имеющие «естественные» уникальные значения.

Правило 3. Функциональная зависимость.

Для каждого значения первичного ключа значения в столбцах данных должно относиться к объекту таблицы и полностью его описывать: 1. в таблице не должно быть данных, не относящихся к объекту, определяемому первичным ключом; 2. данные в таблице должны полностью описывать объект.

Правило 4. Независимость полей

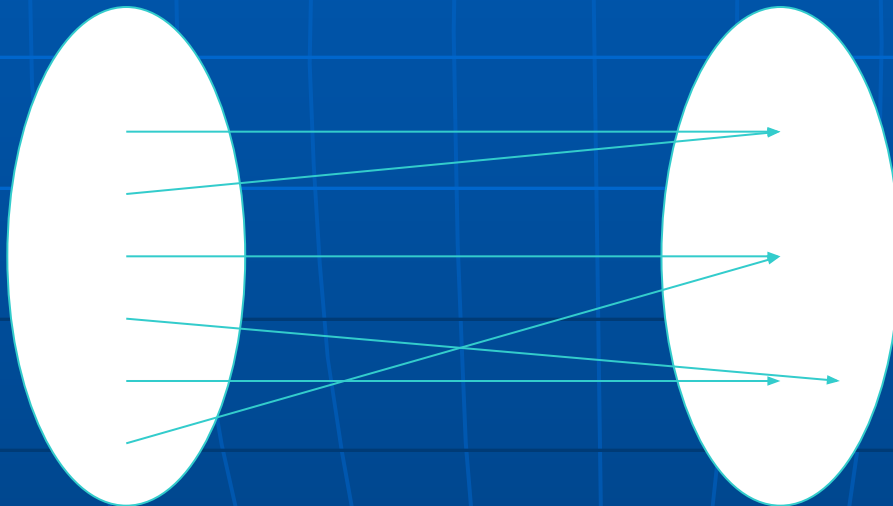
Должна быть возможность изменять значения любого поля (кроме первичного ключа) без воздействия на данные других полей.

Отображение «Многие к одному»

**Отображение имеет тип *Многие к одному*,
если оно является функцией**

Аргумент

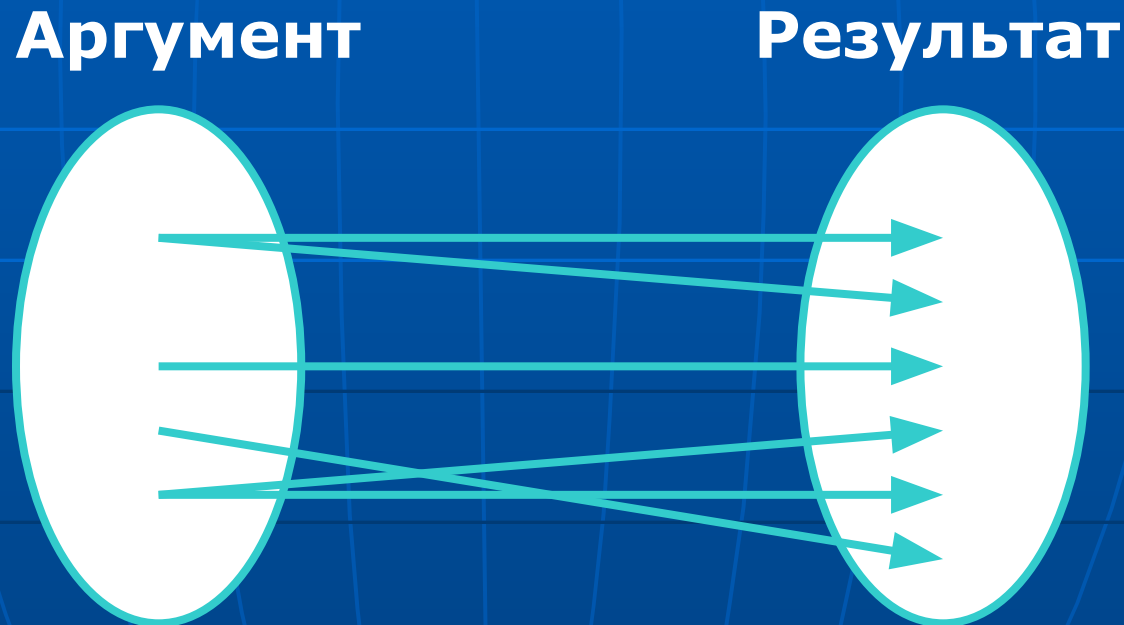
Результат



Из значения аргумента выходит одна стрелка

Отображение «Один ко многим»

Отображение имеет тип *Один ко многим*, если для каждого значения результата отображения имеется только одно значение аргумента. При этом одно значение аргумента может отображаться в несколько значений результата



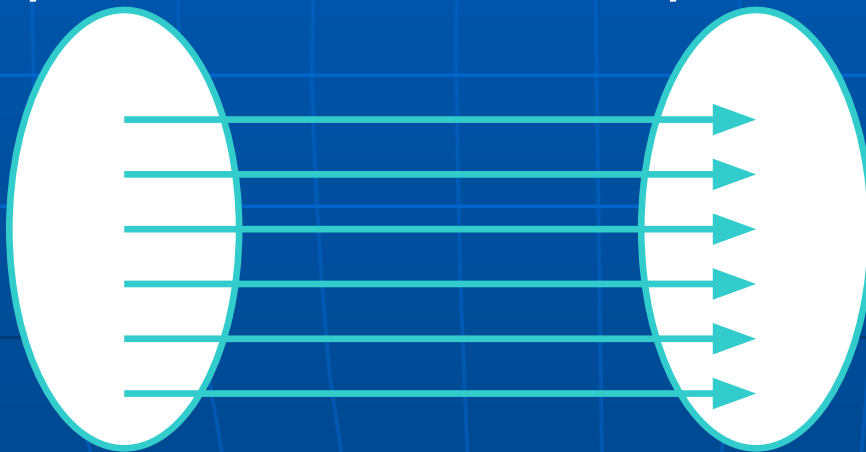
В каждое значение результата входит одна стрелка

Отображение «один к одному»

Отображение имеет тип **Один к одному**, если каждому значению аргумента соответствует одно значение результата и наоборот.

Аргумент

Результат



Из каждого значения аргумента выходит ровно одна стрелка и в каждое значение результата входит тоже ровно одна стрелка

Целостность данных

Правила, обеспечивающие поддержание установленных межтабличных связей при вводе или удалении записей. Если наложены условия целостности данных, Access не позволяет добавлять в связанную таблицу записи, для которых нет соответствующей записей в главной таблице, или же изменять записи в главной таблице таким образом, что после этого в связанной таблице появятся записи, не имеющие соответствующих главных записей, а также удалять записи в главной таблице, для которых имеются подчиненные записи в связанной таблице».

Параметры целостности:

Обеспечение целостности

Этот параметр устанавливается только в том случае, если соответствующее поле главной таблицы является первичным ключом, связанные поля имеют один и тот же тип данных или обе таблицы содержатся в одной БД

Каскадное обновление связанных полей:

Для автоматического обновления соответствующих значений в связанной таблице при изменении значения ключевого поля в базовой таблице

Каскадное удаление связанных записей:

Для автоматического удаления связанных записей в связанной таблице при удалении записи в базовой таблице

Главная таблица: типы связи

Если оба связываемых атрибута не являются ключевыми, то главной будет таблица, от которой пользователь начинает протягивать связь. В этом случае тип связи не устанавливается.

Главная таблица: типы связи

Если в связываемых таблицах ровно один из двух связываемых атрибутов объявлен ключевым, то главной будет та таблица, к которой относится ключевой атрибут. В этом случае при установлении обеспечения целостности данных получается связь типа «один ко многим».

Главная таблица: типы связи

Если в связываемых таблицах оба связываемых атрибута объявлены ключевыми, то главная таблица назначается пользователем (протягиванием в нужном направлении связи между атрибутами). При установлении обеспечения целостности данных получается связь типа «один к одному»

СВЯЗЬ

«Связь, это пара таблиц, в каждой из которых выделено по набору атрибутов, с указанием типа соединения и параметров целостности» А. Г. Гейн

СВОЙСТВО СВЯЗИ

1. Если связь имеет тип «один к одному», то каждая строка главной таблицы связана не более чем с одной строкой подчиненной таблицы и каждая строка подчиненной таблицы связана в точности с одной строкой главной таблицы.

СВОЙСТВО СВЯЗИ

2. Если связь имеет тип «один ко многим», то каждая строка подчиненной таблицы связана в точности с одной строкой главной таблицы, но каждая строка главной таблицы может быть связана с несколькими строками подчиненной таблицы.

СВОЙСТВО СВЯЗИ

3. Если одной записи в главной таблице могут соответствовать несколько записей связанной таблицы, и наоборот, одной записи в подчиненной таблице могут соответствовать несколько записей главной таблицы, такая запись называется «**многие ко многим**». Две таблицы, находящиеся в отношении «**многие ко многим**» могут быть связаны с помощью третьей (промежуточной) таблицы, в которой присутствуют по одному атрибуту, в точности повторяющие один из атрибутов связанной и главной таблицы. Промежуточная таблица должна быть связана с двумя другими таблицами по данным атрибутам связью «**один к одному**» или «**один ко многим**».

Форма

«Форма – это совокупность полей, кнопок и других элементов управления, расположенных на одной или нескольких страницах. Применяется в основном для ввода, отображения, корректировки и поиска данных.» А. Г. Гейн.

Поле – клетка таблицы. Прямоугольник, в котором отображается значение атрибута.

Элемент управления – графический объект, например, поле, флажок, прямоугольник или кнопка, размещаемый пользователем в форме или отчете в режиме конструктора для отображения данных, выполнения макрокоманд или упрощения чтения формы или отчета.