

# Реляционные базы данных. Целостность данных

**Целостность** (от англ. integrity – нетронутость, неприкосновенность, сохранность, целостность) – понимается как правильность данных в любой момент времени.

Поддержание целостности базы данных может рассматриваться как защита данных от неверных изменений или разрушений

**Целостность данных - это механизм поддержания соответствия базы данных предметной области.**

В реляционной модели данных определены два базовых требования обеспечения целостности:

- **целостность сущностей**
- **целостность ссылок**

## Целостность сущностей

Объект реального мира представляется в реляционной базе данных как кортеж некоторого отношения (запись таблицы).

Требование целостности сущностей заключается в следующем: **каждый кортеж любого отношения должен отличаться от любого другого кортежа этого отношения (т.е. любое отношение должно обладать первичным ключом).**

Вполне очевидно, что если данное требование не соблюдается (т.е. кортежи в рамках одного отношения не уникальны), то в базе данных может храниться противоречивая информация об одном и том же объекте.

## Целостность сущностей

Поддержание целостности сущностей обеспечивается средствами системы управления базой данных (СУБД).

Это осуществляется с помощью двух ограничений:

- при добавлении записей в таблицу проверяется уникальность их первичных ключей;
- не допускается изменение значений атрибутов, входящих в первичный ключ.

**Правило целостности сущностей:**

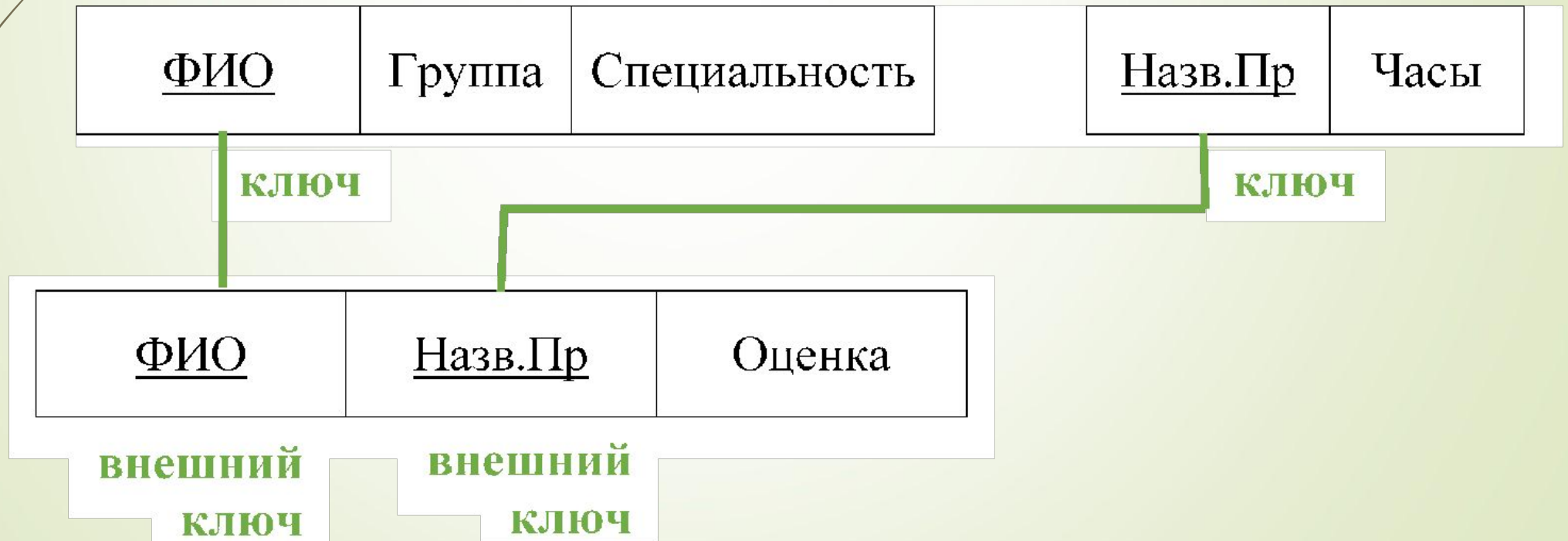
**Атрибуты, входящие в состав некоторого потенциального ключа не могут принимать null-значений.**

# Реляционные базы данных. Целостность данных

## Целостность ссылок

Сложные объекты реального мира представляются в реляционной базе данных в виде нескольких нормализованных отношений, связанных между собой.

Пусть в отношении **R1** имеется не ключевой атрибут **A**, значения которого являются значениями ключевого атрибута **B** другого отношения **R2**. Тогда атрибут **A** отношения **R1** есть внешний ключ.



# Реляционные базы данных. Целостность данных

## Связывание таблиц реализует:

- ✓ контроль целостности вводимых в базу данных,
- ✓ повышение достоверности хранимой в БД информации,
- ✓ облегчение доступа к данным.

При связывании двух таблиц выделяют **основную** и **дополнительную (подчиненную)** таблицы.

Логическое связывание таблиц производится с помощью **ключа связи**.

Ключ связи состоит из одного или нескольких полей - **полей связи**

Для отражения функциональных зависимостей между отношениями используется дублирование первичного ключа одного отношения (родительского) в другое (дочернее).

**Правило целостности внешних ключей.**

**Для каждого значения внешнего ключа, появляющегося в дочернем отношении, в родительском отношении должен найтись кортеж с таким же значением первичного ключа.**

# Реляционные базы данных. Связи между таблицами

- ✓ один — один (1:1);
- ✓ один — много (1:M);
- ✓ много — один (M:1);
- ✓ много — много (M:M или M:N).

## Характеристика видов связей таблиц

Характеристика полей связи по видам	1:1	1:M	M:1	M:M
Поля связи основной таблицы	являются ключом	являются ключом	не являются ключом	не являются ключом
Поля связи дополнительной таблицы	являются ключом	не являются ключом	являются ключом	не являются ключом

# Реляционные базы данных. Связи между таблицами

**Пример.** Дана совокупность информационных объектов, отражающих учебный процесс в вузе:

- ✓ СТУДЕНТ (Номер, Фамилия, Имя, Отчество, Пол, Дата рождения, Группа)
- ✓ СЕССИЯ (Номер, Оценка1, Оценка2, Оценка3, Оценка4, Результат)
- ✓ СТИПЕНДИЯ (Результат, Процент)
- ✓ ПРЕПОДАВАТЕЛЬ (Код преподавателя, Фамилия, Имя, Отчество).

## Связь вида 1:1

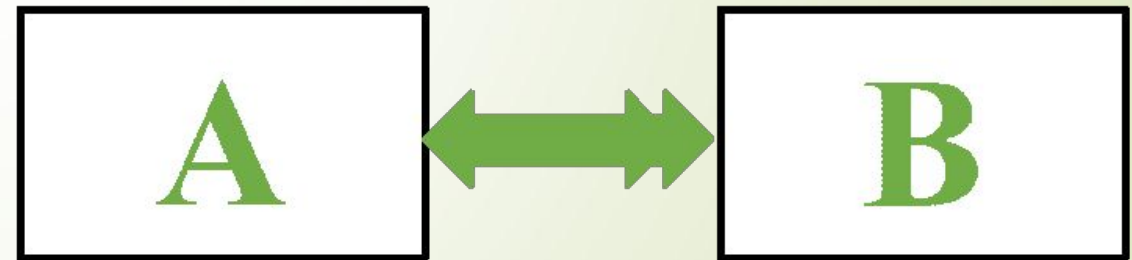
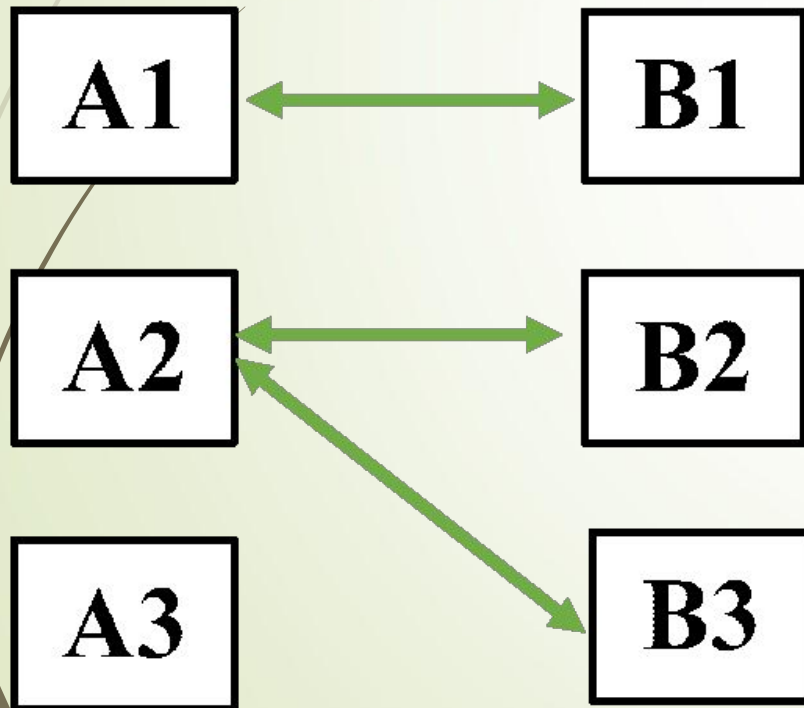
Предполагает, что в каждый момент времени одному экземпляру информационного объекта **A** соответствует не более одного экземпляра информационного объекта **B** и наоборот.

Связь вида 1:1 образуется в случае, когда все поля связи основной и дополнительной таблиц являются ключевыми. Поскольку значения в ключевых полях обеих таблиц не повторяются, обеспечивается взаимно-однозначное соответствие записей из этих таблиц. Сами таблицы, по сути, здесь становятся



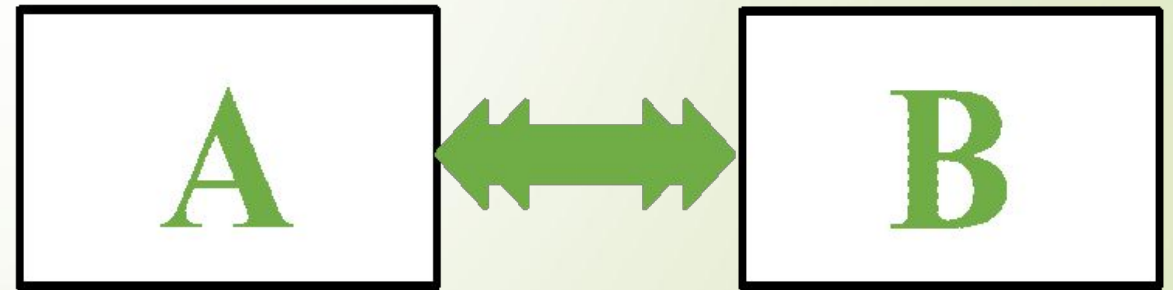
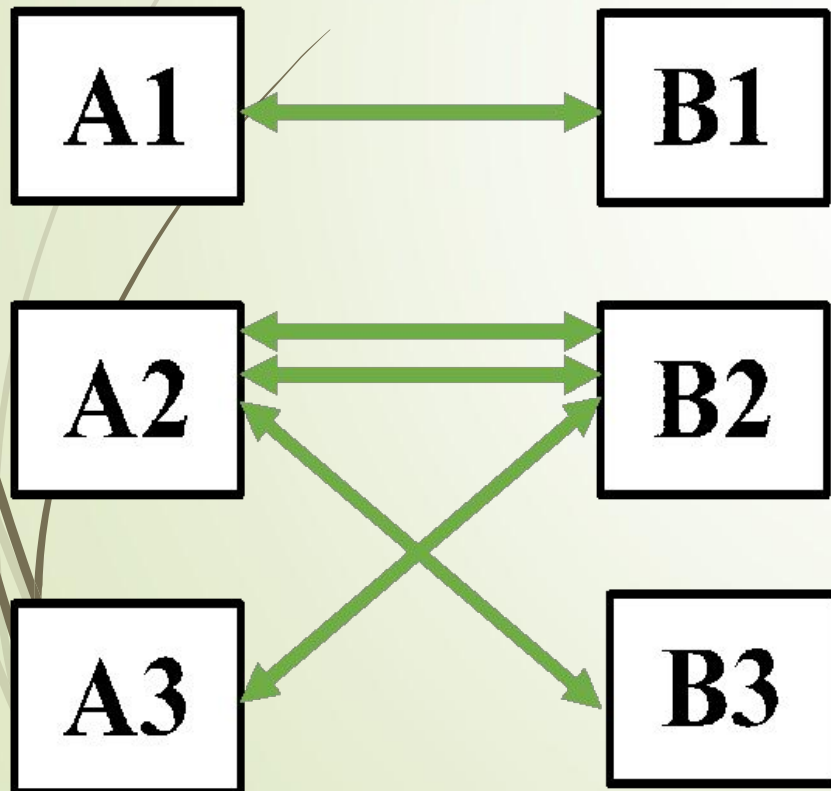
## Связь вида 1:М (М:1)

Предполагает, что в каждый момент времени одному экземпляру информационного объекта **A** соответствует 0,1 или более одного экземпляра информационного объекта **B** и наоборот.



## Связь вида М:М

Предполагает, что в каждый момент времени одному экземпляру информационного объекта **A** соответствует 0, 1 или более экземпляров объекта **B** и наоборот.



## Контроль целостности связей

Каждой записи основной таблицы соответствует NULL или более записей дополнительной таблицы.

В дополнительной таблице нет записей, которые не имеют родительских записей в основной таблице.

Каждая запись дополнительной таблицы имеет только одну родительскую запись основной таблицы.

## Операции

### **для родительского отношения**

#### **1. Вставка кортежа в родительском отношении.**

При вставке кортежа в родительское отношение возникает новое значение потенциального ключа. Т.к. допустимо существование кортежей в родительском отношении, на которые нет ссылок из дочернего отношения, то вставка кортежей в родительское отношение не нарушает ссылочной целостности.

#### **2. Обновление кортежа в родительском отношении.**

При обновлении кортежа в родительском отношении может измениться значение потенциального ключа, что может привести к нарушению ссылочной целостности, если это обновление затрагивает значение потенциального ключа.

## Операции для родительского отношения

### 3. Удаление кортежа в родительском отношении.

При удалении кортежа в родительском отношении удаляется значение потенциального ключа. Если есть кортежи в дочернем отношении, ссылающиеся на удаляемый кортеж, то значения их внешних ключей станут некорректными.

## Операции для дочернего отношения

### 1. Вставка кортежа в дочернее отношение.

Нельзя вставить кортеж в дочернее отношение, если вставляемое значение внешнего ключа некорректно. Вставка кортежа в дочернее отношение может привести к нарушению ссылочной целостности

## Операции **для дочернего отношения**

### **2. Обновление кортежа в дочернем отношении.**

При обновлении кортежа в дочернем отношении можно попытаться некорректно изменить значение внешнего ключа. Обновление кортежа в дочернем отношении может привести к нарушению ссылочной целостности.

### **3. Удаление кортежа в дочернем отношении.**

При удалении кортежа в дочернем отношении ссылочная целостность не нарушается.

**Ссылочная целостность** может быть нарушена при выполнении одной из **четырёх** операций:

1. Обновление кортежа в родительском отношении.
2. Удаление кортежа в родительском отношении.
3. Вставка кортежа в дочернее отношение.
4. Обновление кортежа в дочернем отношении.

## Основные стратегии поддержания ссылочной целостности:

1. **RESTRICT (ОГРАНИЧИТЬ)** - не разрешать выполнение операции, приводящей к нарушению ссылочной целостности.
2. **CASCADE (КАСКАДИРОВАТЬ)** - разрешить выполнение требуемой операции, но внести при этом необходимые поправки в других отношениях так, чтобы не допустить нарушения ссылочной целостности и сохранить все имеющиеся связи. Изменение начинается в родительском отношении и каскадно выполняется в дочернем отношении.



## Дополнительные стратегии поддержания ссылочной целостности:

1. **SET NULL (УСТАНОВИТЬ В NULL)** - разрешить выполнение требуемой операции, но все возникающие некорректные значения внешних ключей изменять на null-значения.
2. **SET DEFAULT (УСТАНОВИТЬ ПО УМОЛЧАНИЮ)** - разрешить выполнение требуемой операции, но все возникающие некорректные значения внешних ключей изменять на некоторое значение, принятое по умолчанию.
3. **IGNORE (ИГНОРИРОВАТЬ)** - выполнять операции, не обращая внимания на нарушения ссылочной целостности.

# Реляционные базы данных. Связи между таблицами

При вводе новых записей данные сначала вводят в основную таблицу, а потом – в дополнительную.

## При модификации записи:

- ✓ запись дополнительной таблицы может сменить родителя, но остаться без него не должна;
- ✓ новое значение поля связи дополнительной таблицы должно совпадать с соответствующим значением какой-либо записи в основной таблице.

# Реляционные базы данных. Связи между таблицами

## Правила редактирования поля связи основной таблицы:

- ✓ Редактируются записи, у которых нет подчиненных записей. Если есть подчиненные записи, то запрет модификации полей связи.
- ✓ Изменение в полях связи основной таблицы можно, при мгновенной передаче во все поля связи всех записей дополнительной таблицы (каскадирование).

Удаление записи. Удаление записи основной таблицы необходимо подчинить следующим правилам:

1. Можно удалять запись, не имеющую подчиненных записей.
2. Запретить удаление записей при наличии подзаписей или удаление ее со всеми подчиненными записями (каскадное удаление).