



Тема 4.
**Инфолингвистическое
(концептуальное)
моделирование предметной
области**






Лекция 4

Вопросы лекции:

1. Концептуальное проектирование базы данных
2. Создание диаграммы «сущность-связь»
3. Пример разработки ER-модели

1. Концептуальное проектирование базы данных

Концептуальная модель:

-  Называется «**Модель сущность-связь**»,
-  Иллюстрируется с использованием «**Диаграммы сущность-связь**» (ERD)
-  Является результатом завершения процесса моделирования данных
-  Предприятия используют данные для увеличения продаж и/или снижения затрат.
-  Чтобы точно собирать эти данные, бизнес должен создать концептуальную модель данных, которые он считает важными.

Концептуальная модель важна для бизнеса, потому что она:

- Описывает именно информационные потребности бизнеса
- Облегчает дискуссию (коммуникации)
- Предотвращает ошибки и недоразумения
- Формирует важную документацию «идеальной системы»
- Формирует надежную основу для проектирования физической базы данных
- Документирует процессы (также известные как «бизнес-правила») бизнеса
- Принимает во внимание правила и законы, регулирующие предметную область.

Сущность:

- «Что-то», что имеет значение для предметной области, о которой должны быть известны данные
- Имя для набора (класса) похожих объектов, которые вы можете перечислить
- Обычно существительное
- Примеры: **объекты, события, люди**
- У сущностей есть **экземпляры**.
- Экземпляр** - это единственное вхождение объекта.

Назначение сущностей

- Зная, как организовать и классифицировать данные, можно сделать полезные выводы о кажущихся случайными фактах.
- Наш богатый технологиями мир производит огромное количество фактов, нуждающихся в структуре и порядке.
- Важно знать о сущностях, потому что это то, о чем мы храним данные.

Например:

Университет должен хранить данные о (как минимум): СТУДЕНТАХ, ПРЕПОДАВАТЕЛЯХ, ДИСЦИПЛИНАХ, ГРУППАХ, АУДИТОРИЯХ и др.

Назначение атрибутов

- ▣ **Атрибуты** предоставляют более конкретную информацию об объекте.
- ▣ Атрибуты помогают различать экземпляры сущности, предоставляя для нее более подробные сведения.

Например:

При создании нескольких отчетов о продажах вы должны иметь возможность идентифицировать конкретный отчет из списка отчетов.

Назначение уникальных идентификаторов

- Уникальные идентификаторы позволяют отличить один экземпляр объекта от другого.

Например:








Отличить одного студента от другого.

При перечислении транзакций в финансовом отчете отличать одну транзакцию от другой.

Сущности и экземпляры

Сущность	Экземпляры
Факультет	Менеджмента, ФМЭСИ, Маркетинга ...
Кафедра	Базовая кафедра компании 1С Высшей математики ПИиИБ ...

Связи в моделях данных

-  Показывают, как объекты связаны друг с другом
-  Существуют только между объектами (или одним объектом и самим собой)
-  Двухнаправленные
-  Названы с обоих концов
-  Имеют опциональность
-  Имеют мощность
-  Имеют свойство трансферабельности

Опциональность (модальность) связи

Связи являются обязательными или необязательными.

Рассмотрим два объекта СОТРУДНИК и РАБОТА. Определим опциональность, ответив на два вопроса:

1. Должен ли каждый иметь работу, т.е. для работника обязательна или необязательна связь с работой?
2. Должна ли каждая работа назначаться сотруднику, т.е. обязательна или необязательна связь работы с сотрудником?

Опциональность (модальность) связи

Модальность связи определяет минимальное значение экземпляров сущностей. Каждая связь может иметь одну из двух модальностей связи:



Модальность **"может"** означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть и не связан ни с одним экземпляром, в этом случае допускаются значения **null** для **внешнего ключа**.

Модальность **"должен"** означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности.

Связь может иметь разную модальность с разных концов.

Мощность в связях

Мощность определяет степень, с которой одна сущность связана с другой, т.е. отвечает на вопрос «Сколько экземпляров сущности связано с экземпляром данной сущности?».

Иначе говоря, **мощность** определяет минимальное и максимальное количество экземпляров сущностей в связи и характеризуется **модальностью связи и типом связи**.

Например:

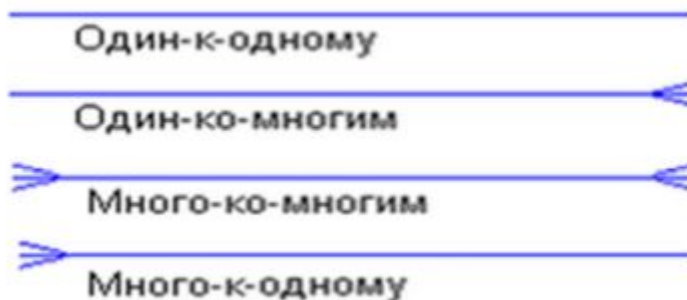
Может ли сотрудник работать в разных структурных подразделениях компании или только в одном (разрешено ли совмещение)?

Сколько сотрудников может выполнять одну конкретную работу? Только один сотрудник? Или более одного сотрудника?

Мощность в связях

Максимальные значения экземпляров сущности для каждого правила отношения, определяется типом связи:

- **связь один-к-одному (1:1);**
- **связь один-ко-многим (1 :M) или многие-к-одному (M: 1);**
- **связь много-ко-многим (M:M);**
- **связь рекурсивная.**



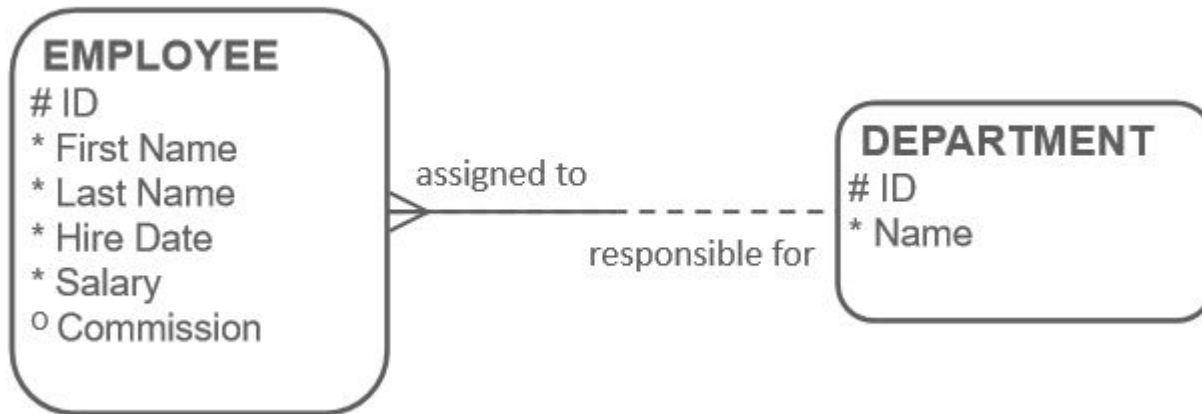
Переносимость связей (трансферабельность)

✍ Трансферабельность определяет возможность изменения родительской записи.

Например:

✍ Может ли СОТРУДНИК быть переведен из одного отдела в другой?

ДА - связь трансферабельна



Переносимость связей (трансферабельность)

✍ Например:

✍ Может ли **СТУДЕНТ** перевестись в другую группу?

ДА - связь трансферабельна



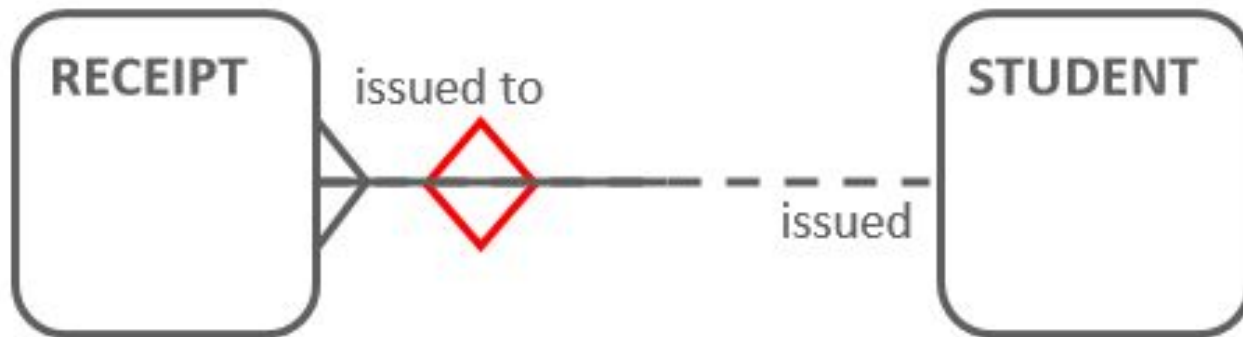
Переносимость связей (трансферабельность)

Например:

✎ **СТУДЕНТУ** может быть выписан **СЧЕТ** за сдачу сертификационного экзамена.



✎ После того, как **СЧЕТ** был выписан, он не может быть передан другому **СТУДЕНТУ**. Если **СЧЕТ** был выписан по ошибке, его нужно отменить, но не передавать другому лицу.

Связь нетрансферабельна




Пример бизнес-сценария 1


Каковы связи в следующем бизнес-сценарии?

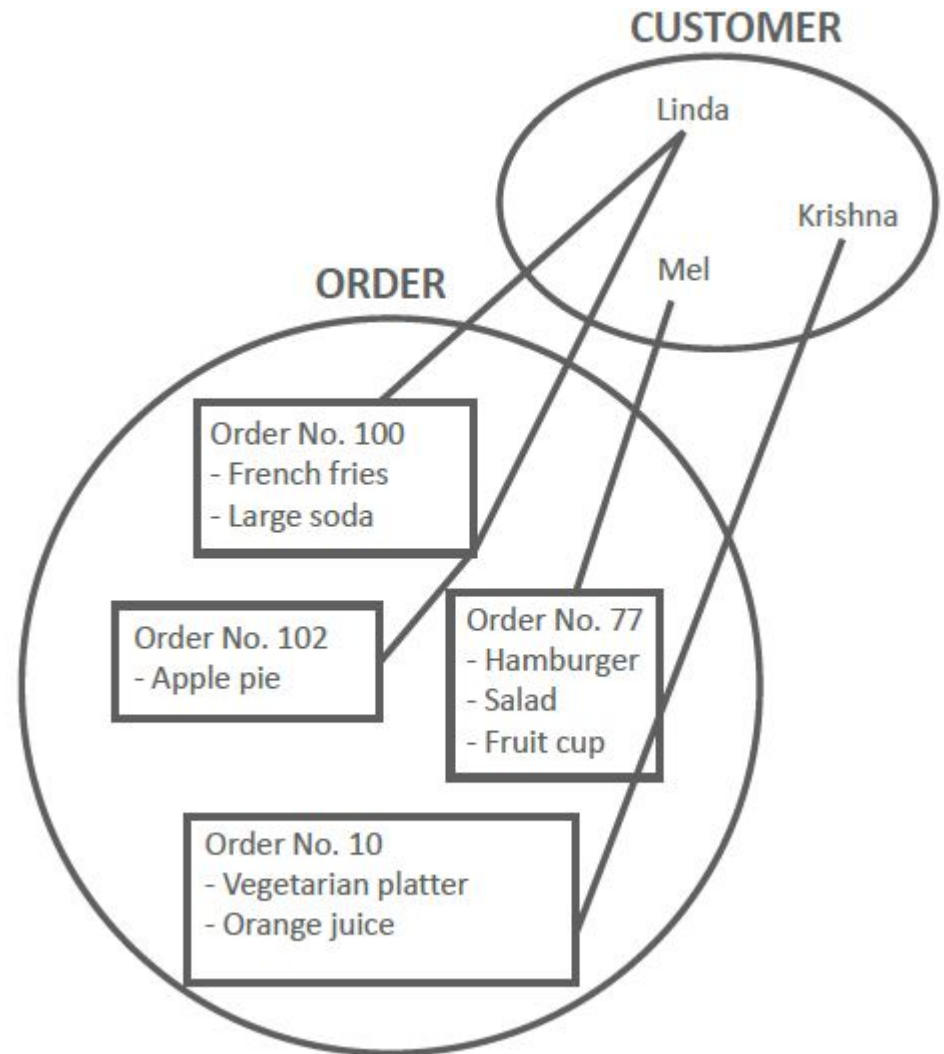
-  В ресторане клиент подходит к стойке и делает заказ. Клиент может заказать не только для себя но и для других. Например, мать заказывает себе и своим детям.
-  Считается, что мать является клиентом, который владеет заказом и несет ответственность за платеж. С течением времени клиент может разместить столько заказов, сколько захочет.

КЛИЕНТЫ ЗАКАЗЫ: ОПЦИОНАЛЬНОСТЬ

 Опциональность =
Должен или может?

 Каждый ЗАКАЗ
(ORDER) должен
быть размещен
одним (и только
одним)
ЗАКАЗЧИКОМ.

 Каждый ЗАКАЗЧИК
(CUSTOMER) должен
разместить один или
несколько ЗАКАЗОВ.

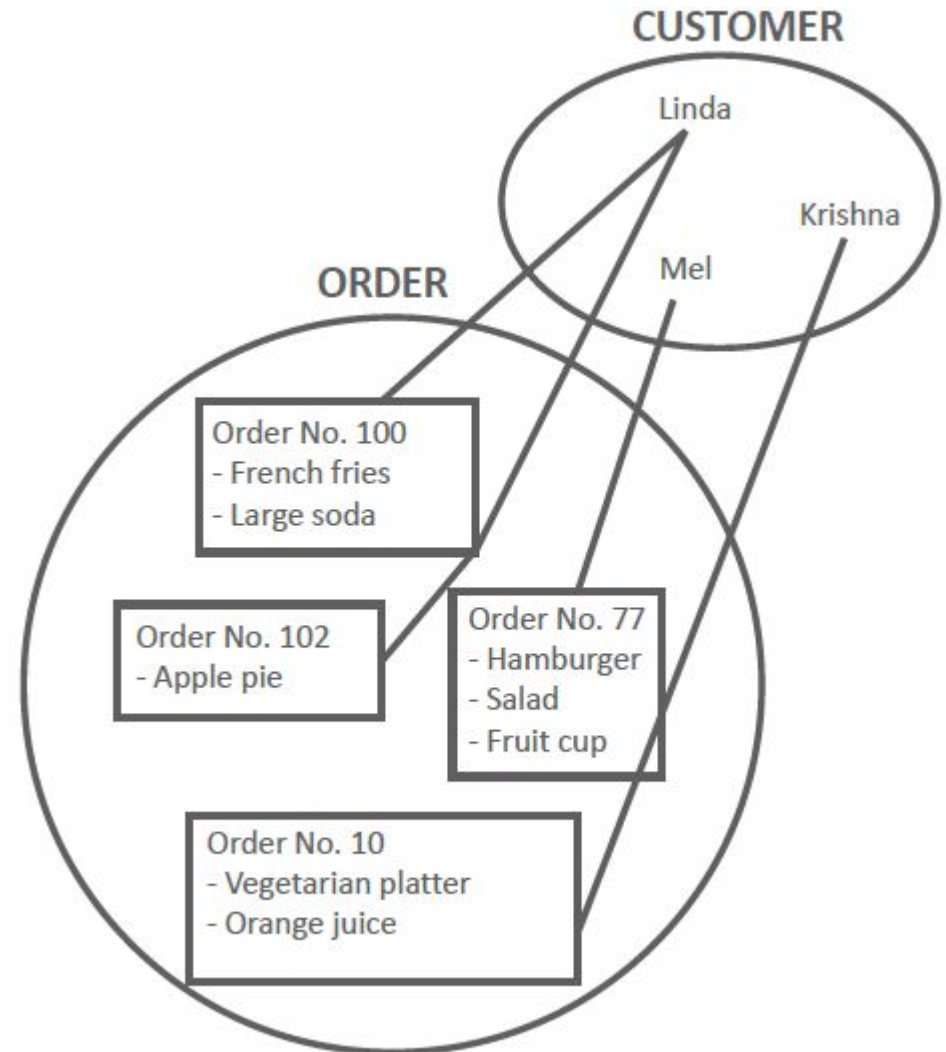


КЛИЕНТЫ ЗАКАЗЫ: МОЩНОСТЬ (кардинальность)

Кардинальность =
Сколько?


Каждый ЗАКАЗ
должен быть
размещен одним и
только одним
ЗАКАЗЧИКОМ.

Каждый ЗАКАЗЧИК
должен разместить
один или несколько
ЗАКАЗОВ.






Пример бизнес-сценария 2

Связи могут присоединяться в сущности к самой себе.

 Необходимо отслеживать сотрудников и их менеджеров. У каждого сотрудника есть один менеджер, в том числе управляющий, который управляет собой. Каждый менеджер может управлять несколькими сотрудниками.

Пример бизнес-сценария 2

Связи могут присоединяться в сущности к самой себе.

-  Поскольку менеджеры также являются сотрудниками, оба они перечислены в одной СУЩНОСТИ: СОТРУДНИК.
-  Каждый СОТРУДНИК может управляться одним и только одним СОТРУДНИКОМ
-  Каждый СОТРУДНИК может управлять одним или несколькими СОТРУДНИКАМИ

Соглашения по рисованию ER-диаграмм

Сущности
представлены
прямоугольниками.

Имена сущностей
входят в
прямоугольник.

Имена сущностей
указываются в
единственном числе
и записываются
прописными
буквами.

DEPARTMENT

```
# * id_department  
* department
```

POSITION

```
# * id_position  
* position  
* min_salary  
* max_salary
```


Условные обозначения на ER-диаграммах

Атрибуты перечислены под именами сущностей.

Обязательные атрибуты отмечены звездочкой: *

Необязательные атрибуты отмечены кружком: ○

Уникальные идентификаторы отмечены символом «хэш»: #

DEPARTMENT

* id_department
* department

POSITION

* id_position
* position
* min_salary
* max_salary

part_time_worker

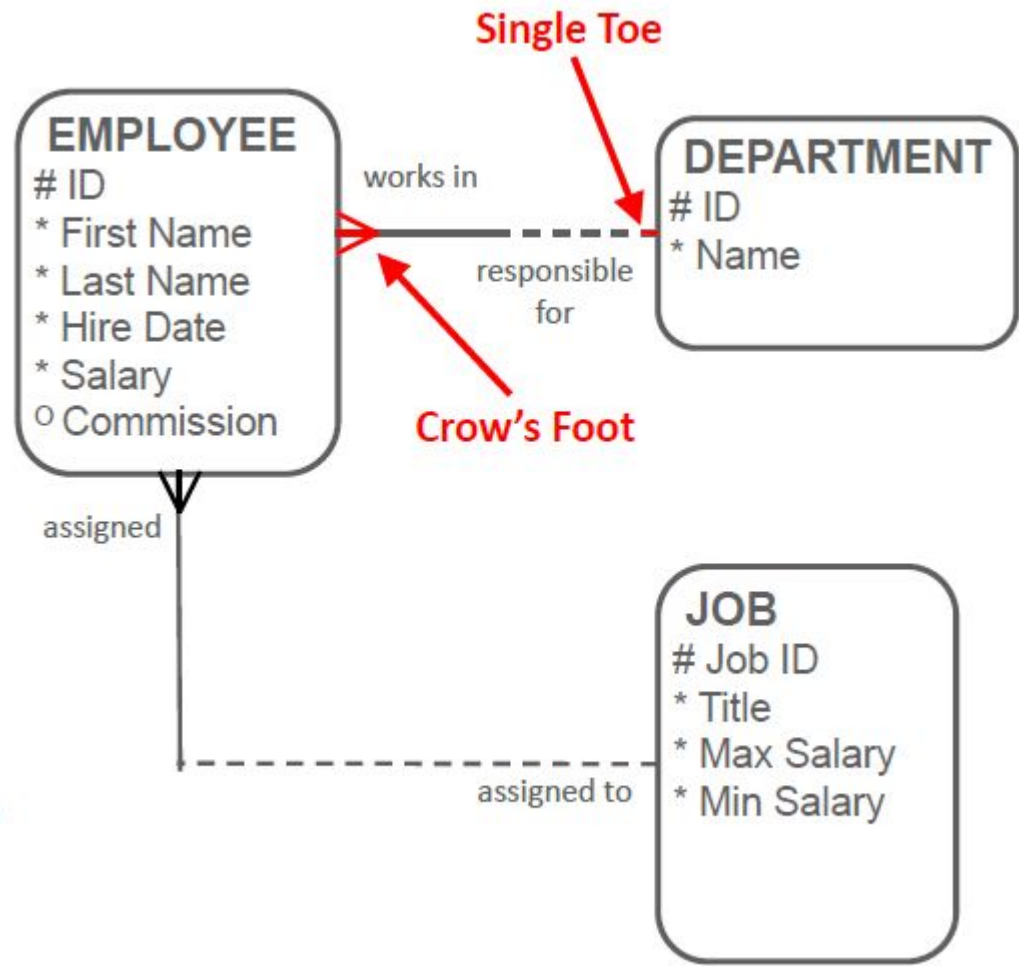
* id_part_time_worker
○ main_job_place

Условные обозначения на ER-диаграммах






Связи - это линии, которые соединяют объекты.

Эти линии являются сплошными или пунктирными.

Эти линии заканчиваются либо одиночной чертой (single toe), либо вороньей лапой (crow's foot).








Как правильно читать ER-диаграмму

-  Каждый экземпляр сущности А
-  ОПЦИОНАЛЬНОСТЬ (должна быть / может быть)
-  Имя связи
-  КАРДИНАЛЬНОСТЬ (один и только один или один или много)
-  Экземпляр сущности В

Как правильно читать ER-диаграмму

Поскольку каждая связь имеет две стороны, читаем связь слева направо (или сверху вниз, в зависимости от схемы ERD).



-  Каждый сотрудник (экземпляр сущности СОТРУДНИК)
-  должен (ОПЦИОНАЛЬНОСТЬ - сплошная линия)
-  работать в (имя связи)
-  одном и только одном (КАРДИНАЛЬНОСТЬ - одиночная черта)
-  отделе (сущность ОТДЕЛ)

Как правильно читать ER-диаграмму

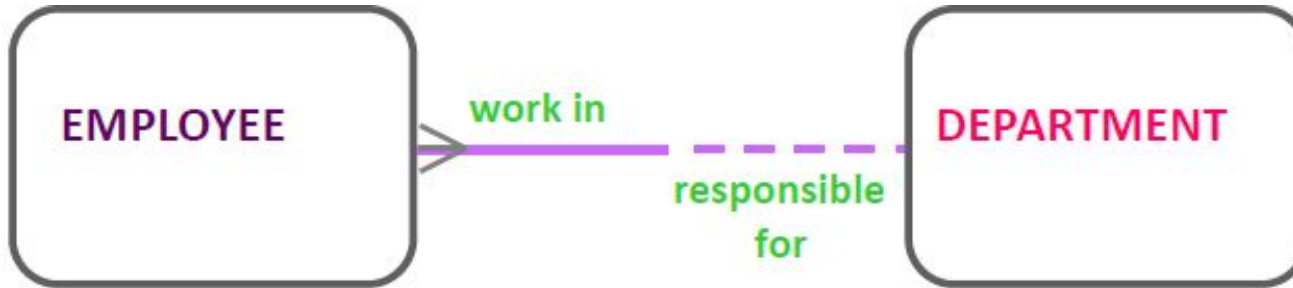
Теперь читаем связь справа налево



- ✍ Каждый отдел (экземпляр сущности ОТДЕЛ)
- ✍ может (ОПЦИОНАЛЬНОСТЬ - пунктирная линия)
- ✍ отвечать за (имя связи)
- ✍ один или более (КАРДИНАЛЬНОСТЬ - воронья лапа)
- ✍ сотрудников (сущность СОТРУДНИК)

Как правильно читать ER-диаграмму

Теперь читаем все вместе



Каждый сотрудник
(экземпляр сущности
СОТРУДНИК)

должен (ОПЦИОНАЛЬНОСТЬ
- сплошная линия)

работать в (имя связи)

одном и только одном
(КАРДИНАЛЬНОСТЬ -
одионочная черта)

отделе (сущность ОТДЕЛ)

Каждый отдел (экземпляр
сущности ОТДЕЛ)

может (ОПЦИОНАЛЬНОСТЬ -
пунктирная линия)

отвечать за (имя связи)

один или более
(КАРДИНАЛЬНОСТЬ - воронья
лапа)

сотрудников (сущность
СОТРУДНИК)

Супертипы и подтипы



Супертипы и подтипы встречаются часто в реальном мире:

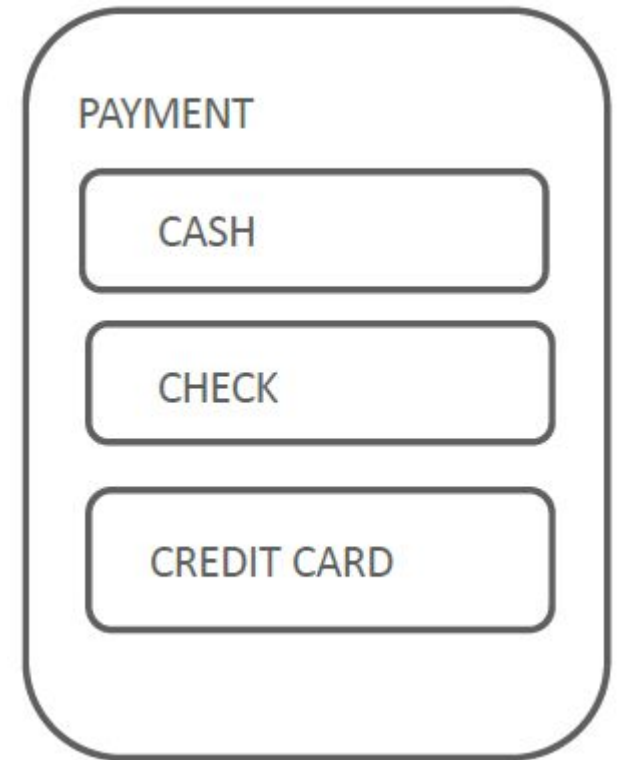
- **ТИПЫ ПИТАНИЯ** (есть, идти)
- **ТИПЫ ПРОДУКТОВЫХ МЕШКОВ** (бумага, пластик)
- **ТИПЫ ПЛАТЕЖЕЙ** (чек, наличные деньги, кредит)
- **ТИПЫ СОТРУДНИКОВ** (штатные, совместители, работающие по договору)

Супертипы и подтипы

Часто некоторые экземпляры объекта имеют атрибуты и / или отношения, которые другие экземпляры не имеют.

Представьте себе бизнес, который должен отслеживать платежи от клиентов.

Клиенты могут оплачивать наличными, чеком или кредитной картой.



Супертипы и подтипы

Все платежи имеют некоторые общие атрибуты: дату платежа, сумму платежа и т. д.

Но только кредитные карты будут иметь атрибут номер карты.

А для оплаты кредитной картой и чеками нам может понадобиться знать, какой **ЗАКАЗЧИК** совершил платеж, в то время как это не нужно для оплаты наличными.

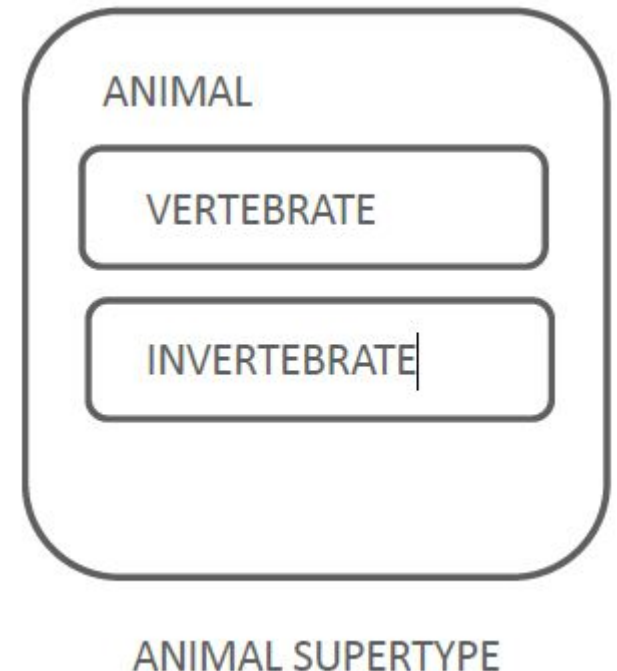
A diagram of a payment form. It consists of a rounded rectangle containing the word "PAYMENT" at the top. Below it are three stacked, rounded rectangular buttons. The top button is labeled "CASH", the middle button is labeled "CHECK", and the bottom button is labeled "CREDIT CARD".

Супертипы и подтипы

Иногда имеет смысл подразделить сущность на подтипы.

Это может иметь место, когда группа экземпляров имеет специальные свойства, такие как атрибуты или отношения, которые существуют только для этой группы.

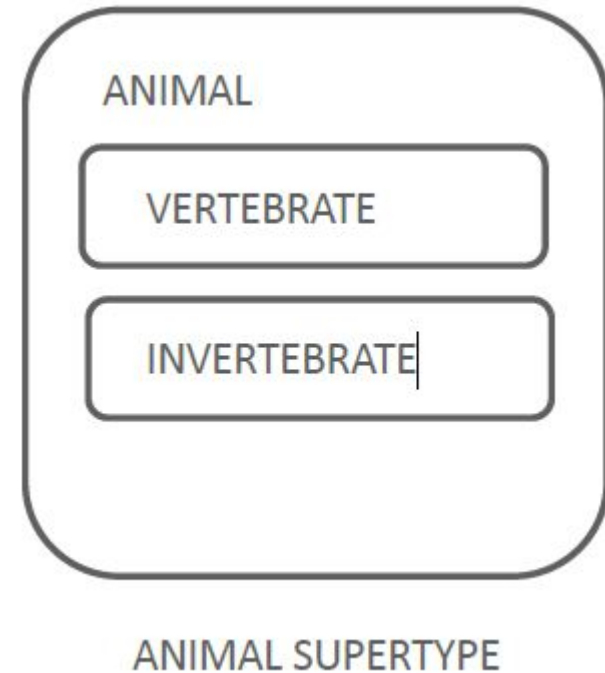
В этом случае объект называется **супертипом**, а каждая группа называется **подтипом**.



Характеристики подтипа

Подтип:

- Наследует все атрибуты супертипа
- Наследует все связи супертипа
- Обычно имеет свои собственные атрибуты или связи
- Изображается внутри супертипа
- Не существует самостоятельно
- Может иметь собственные подтипы



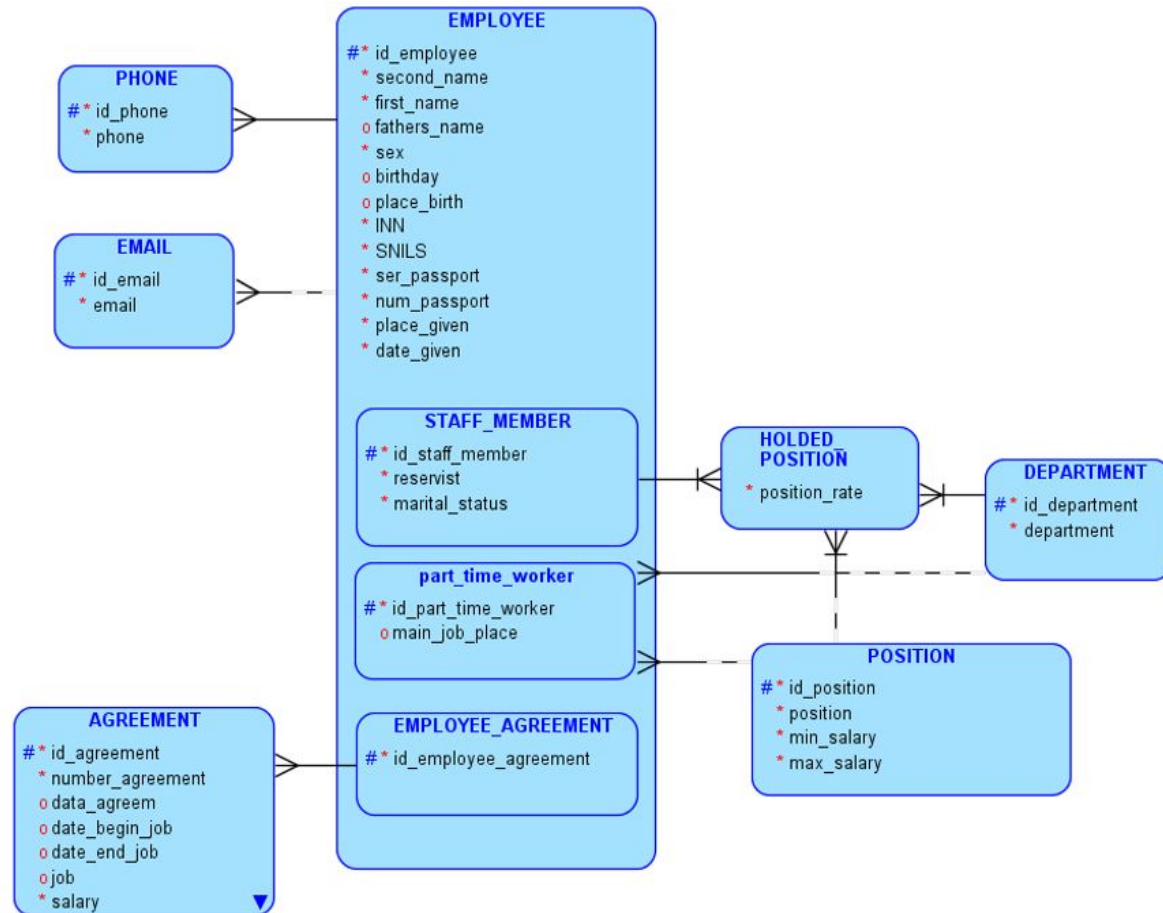
Пример супертипа

Супертип:

EMPLOYEE

Подтипы:

**STAFF_MEMBER,
PART_TIME_WORKER,
EMPLOYEE_AGREEMENT**



Подтипы имеют несколько общих атрибутов.

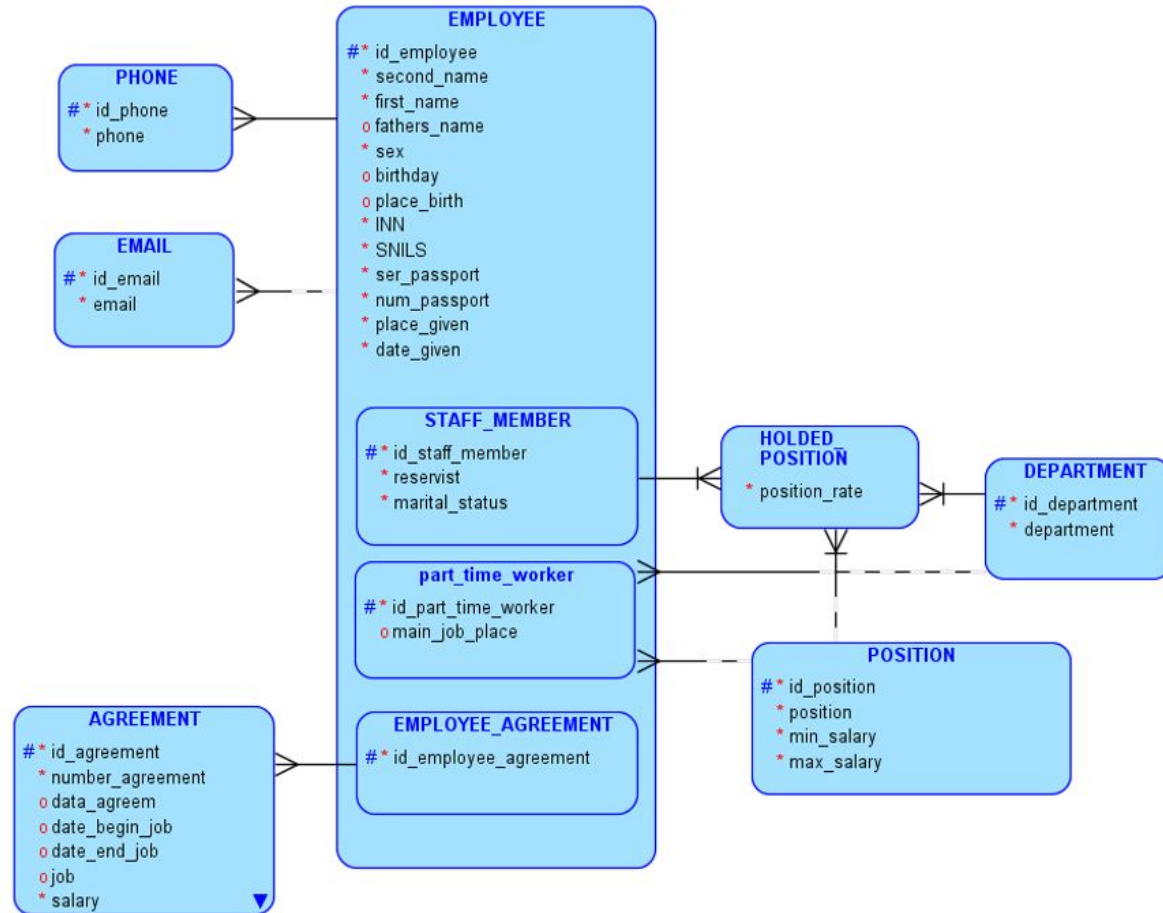
Эти общие атрибуты перечислены на уровне супертипа.

Подтипы наследуют все атрибуты сущности супертипа.

Пример супертипа


То же самое относится к связям.

Связи с сущностями PHONE и EMAIL относятся к супертипу и наследуются подтипами.



Т.о. подтипы наследуют все атрибуты и связи сущности супертипа.

Подтипов должно быть несколько

 Если сущность имеет подтип, должен существовать и второй подтип. Один подтип совпадает с супертипом. Эта идея приводит к двум правилам подтипа:

1. **Исчерпания:** каждый экземпляр супертипа также является экземпляром одного из подтипов. Все подтипы перечислены без упущений.
2. **Исключения:** каждый экземпляр супертипа является экземпляром только одного возможного подтипа.

Подтипов должно быть несколько

На этапе концептуального моделирования хорошей практикой является включение подтипа OTHER, чтобы убедиться, что ваши подтипы являются **исчерпывающими**, - что вы обрабатываете каждый экземпляр супертипа.



Подтипы всегда существуют



Любая сущность может быть декомпозирована на подтипы путем составления правила, которое подразделяет экземпляры на группы.



Однако создавать подтипы имеет смысл только тогда, когда существует потребность показать сходства и различия между экземплярами сущности.

Правильное определение подтипов



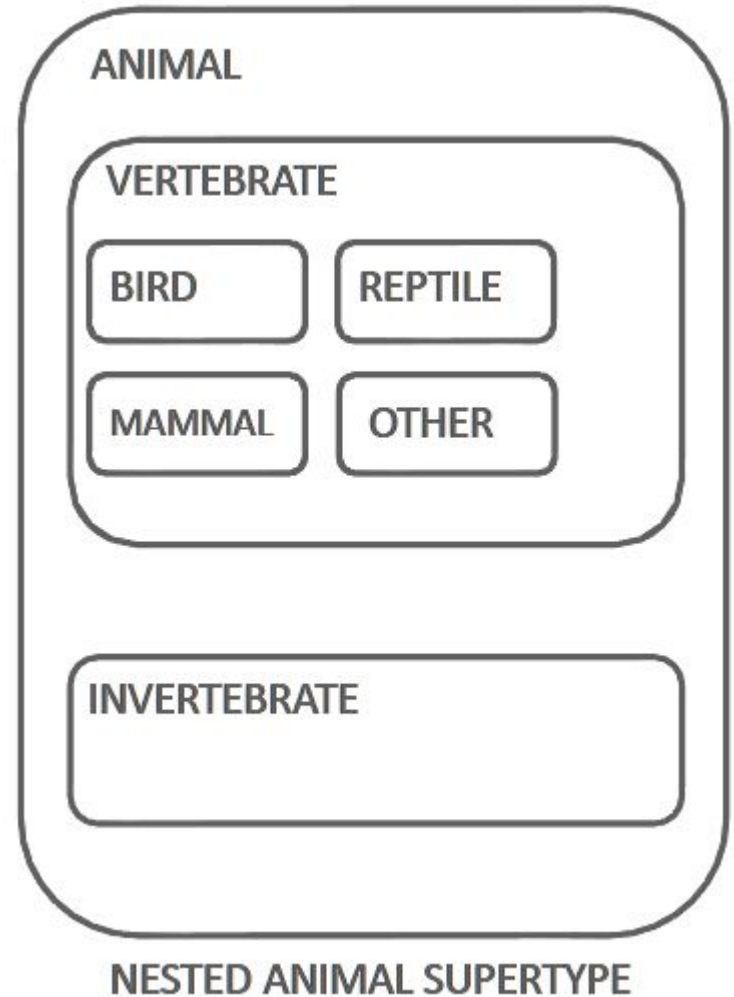
При моделировании супертипов и подтипов следует ответить на три вопроса, чтобы определить, правильно ли идентифицирован подтип:

1. Этот подтип является разновидностью супертипа?
2. Рассмотрены все возможные случаи? (правило **исчерпания**).
3. Каждый экземпляр супертипа принадлежит одному и только одному подтипу? (**исключения**)

Вложенные подтипы

Один подтип можно вложить в другой.

Для удобства чтения обычно показывают подтипы только с двумя уровнями, но нет правила, которое ограничивало бы число уровней вложенности.



2. Документирование бизнес-правил

Структурные и процедурные бизнес-правила

- Структурные бизнес-правила указывают типы информации, подлежащей хранению, и то, как элементы информации взаимосвязаны.
- Процедурные правила касаются предварительных условий, шагов, процессов или требований рабочего процесса для бизнеса.
- Многие процедурные бизнес-правила связаны со временем. Например, событие А должно произойти до события В.
- Структурные бизнес-правила можно практически всегда отображать в ERD.
- Некоторые процедурные бизнес-правила не могут быть отображены на диаграмме, но их необходимо документировать, чтобы их можно было запрограммировать позже.

Примеры структурных бизнес-правил

Структурные бизнес-правила указывают типы информации, подлежащей хранению (атрибуты), и то, как информационные элементы взаимосвязаны (связи).

Вот несколько примеров:

Все заказы в ресторане должны обрабатываться сотрудником (в частности, заказчиком). Система заказа самообслуживания отсутствует.



Примеры структурных бизнес-правил

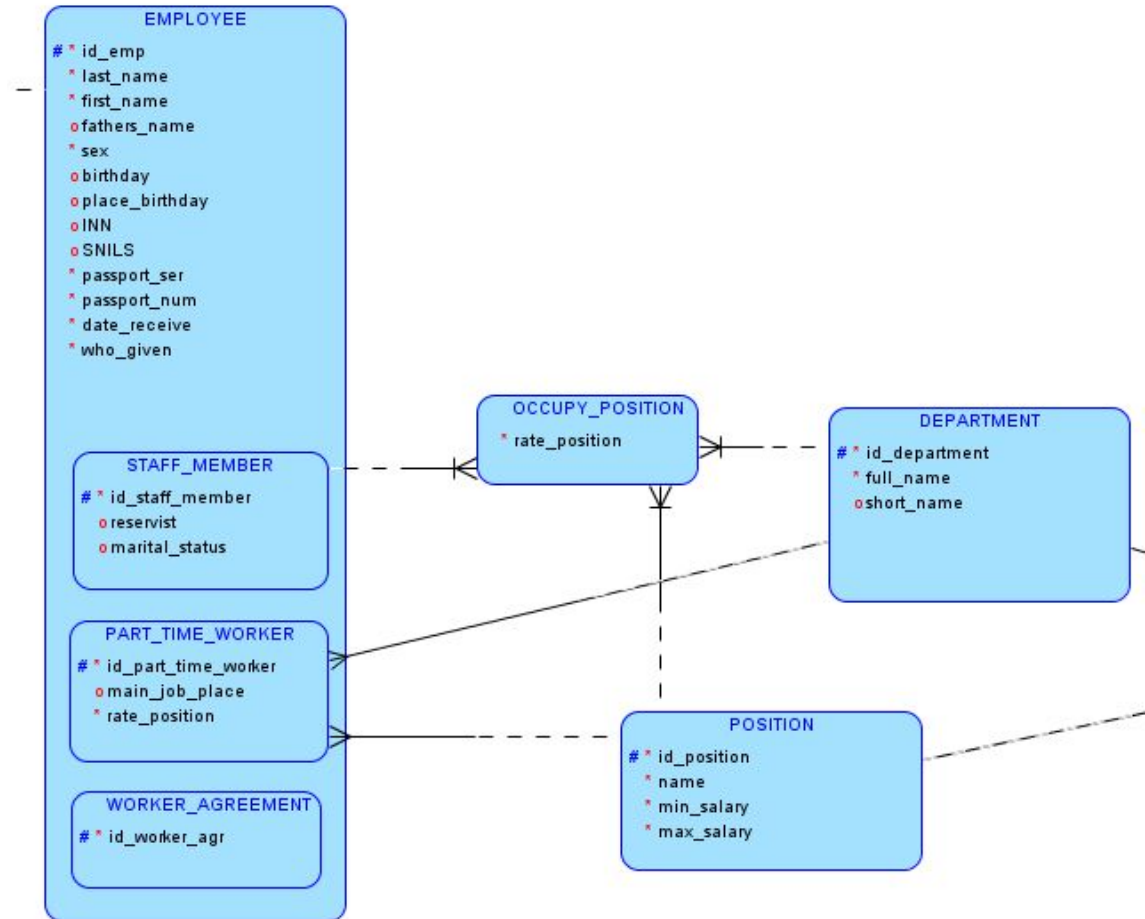
В компании могут работать только сотрудники, имеющие следующий статус:

- ✓ Состоящие в штате;
- ✓ Работающие по совместительству;
- ✓ Работающие на основе договора (ГПХ).

При этом для штатных сотрудников разрешено внутреннее совместительство (в том числе в одном структурном подразделении).

Работающие по совместительству могут занимать только одну должность и состоять только в одном подразделении.

Работающие на основе договора не числятся ни в одном структурном подразделении.



Примеры процедурных бизнес-правил

- ✓ Заявление на служебную командировку может быть подписано руководителем организации только после его подписания непосредственным руководителем сотрудника.
- ✓ Один сотрудник не может одновременно курировать более чем 10 договоров.
- ✓ Студент может выбрать дисциплину (по выбору) CASE-технологии только, если он изучил дисциплину «Проектирование информационных систем»

Невозможность отражения в ER-модели некоторых бизнес-правил

- ✓ В процессе разработки модели концептуальных данных можно моделировать не все бизнес-правила. Некоторые правила, должны быть реализованы путем программирования процессов, которые взаимодействуют с данными.
- ✓ **Например.** Для любого сотрудника, чья сверхурочная работа превышает 10 часов в неделю, часовая тарифная ставка должна быть увеличена в 1,5 раза
- ✓ Клиенты, нарушившие график платежей лишаются скидок.

Спасибо за внимание!