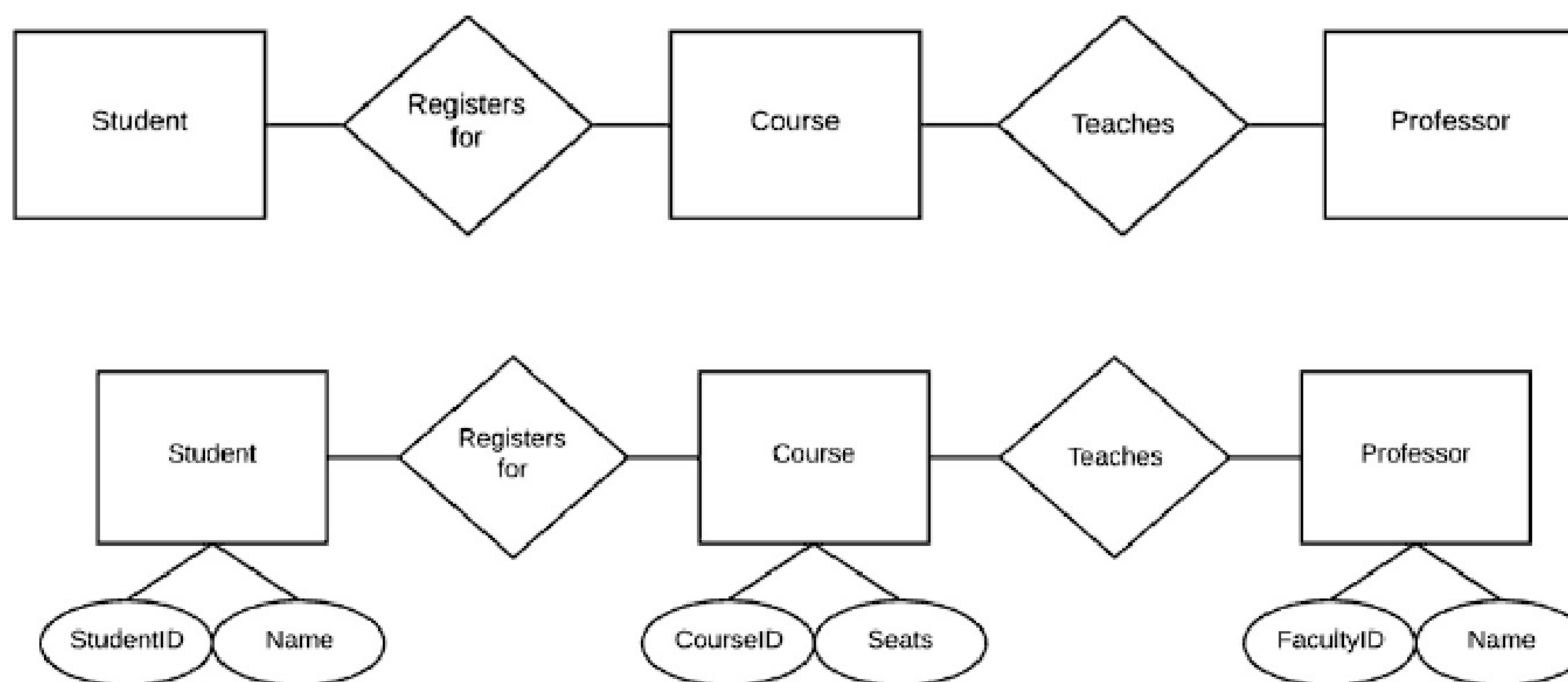
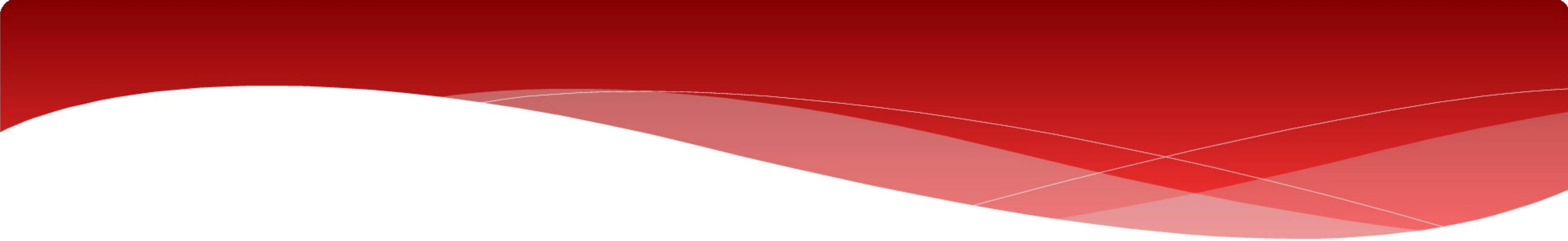


# Построение ER-диаграмм

**Концептуальная модель данных** — схема наивысшего уровня с минимальным количеством подробностей. Достоинство этого подхода заключается в возможности отобразить общую структуру модели и всю архитектуру системы. Менее масштабные системы могут обойтись и без этой модели. В этом случае можно сразу переходить к логической модели.



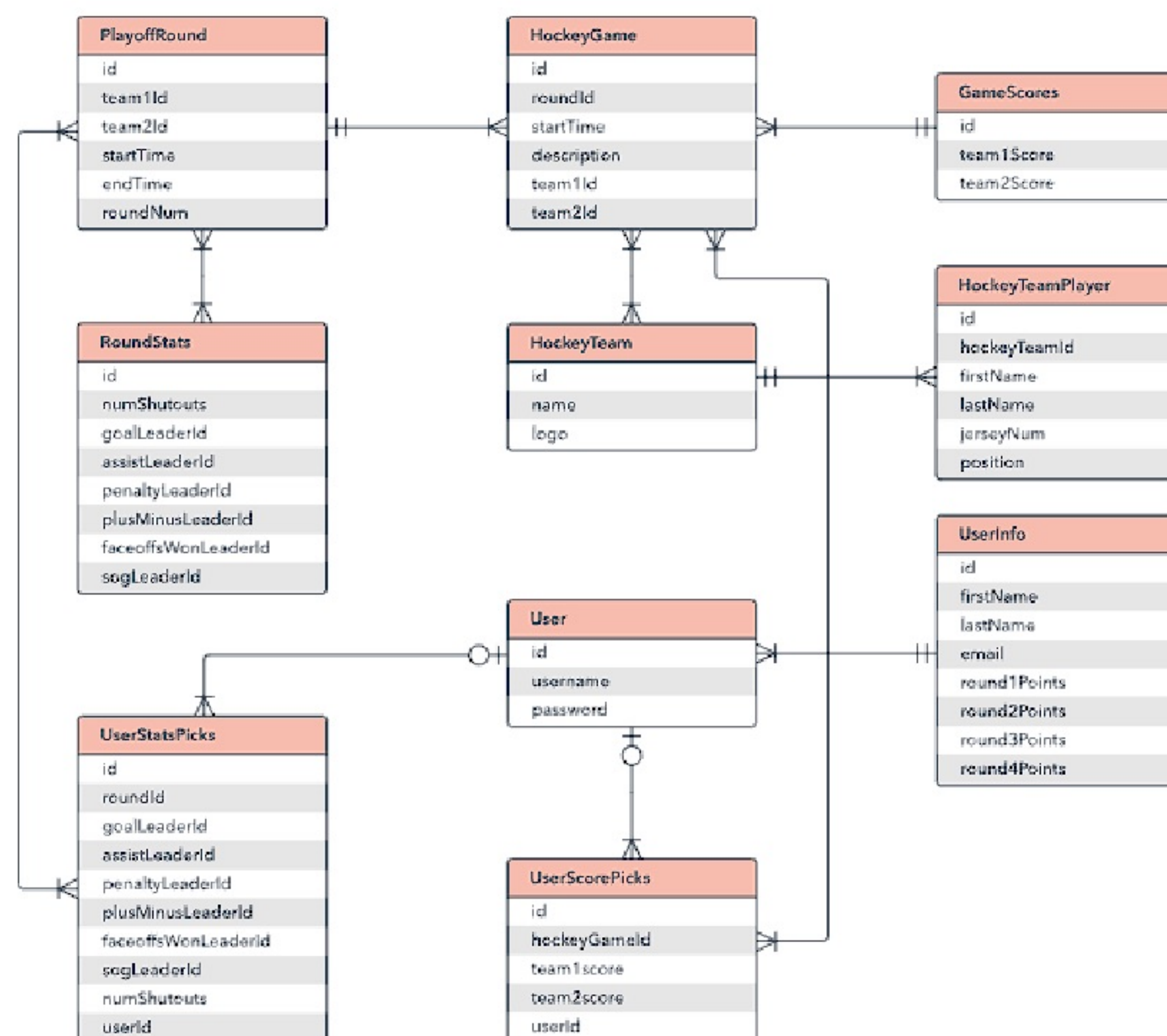


- 
- **Проектирование баз данных.** ER-диаграммы применяются для моделирования и проектирования реляционных баз данных, причем как в плане логических и бизнес-правил (логические модели данных), так и в плане внедрения конкретных технологий (физические модели данных).
  - **Отладка баз данных.** ER-диаграммы применяются для анализа уже имеющихся баз данных с целью выявить и устранить ошибки в логике или развертывании.
  - **Информационные системы для бизнеса.** ER-схемы используются для проектирования и анализа реляционных баз данных, применяемых в бизнес-процессах.



**Логическая модель данных** содержит более подробную информацию, нежели концептуальная модель. На этом уровне определяются более подробные операционные и транзакционные сущности. Логическая модель не зависит от технологии, в которой она будет применяться.

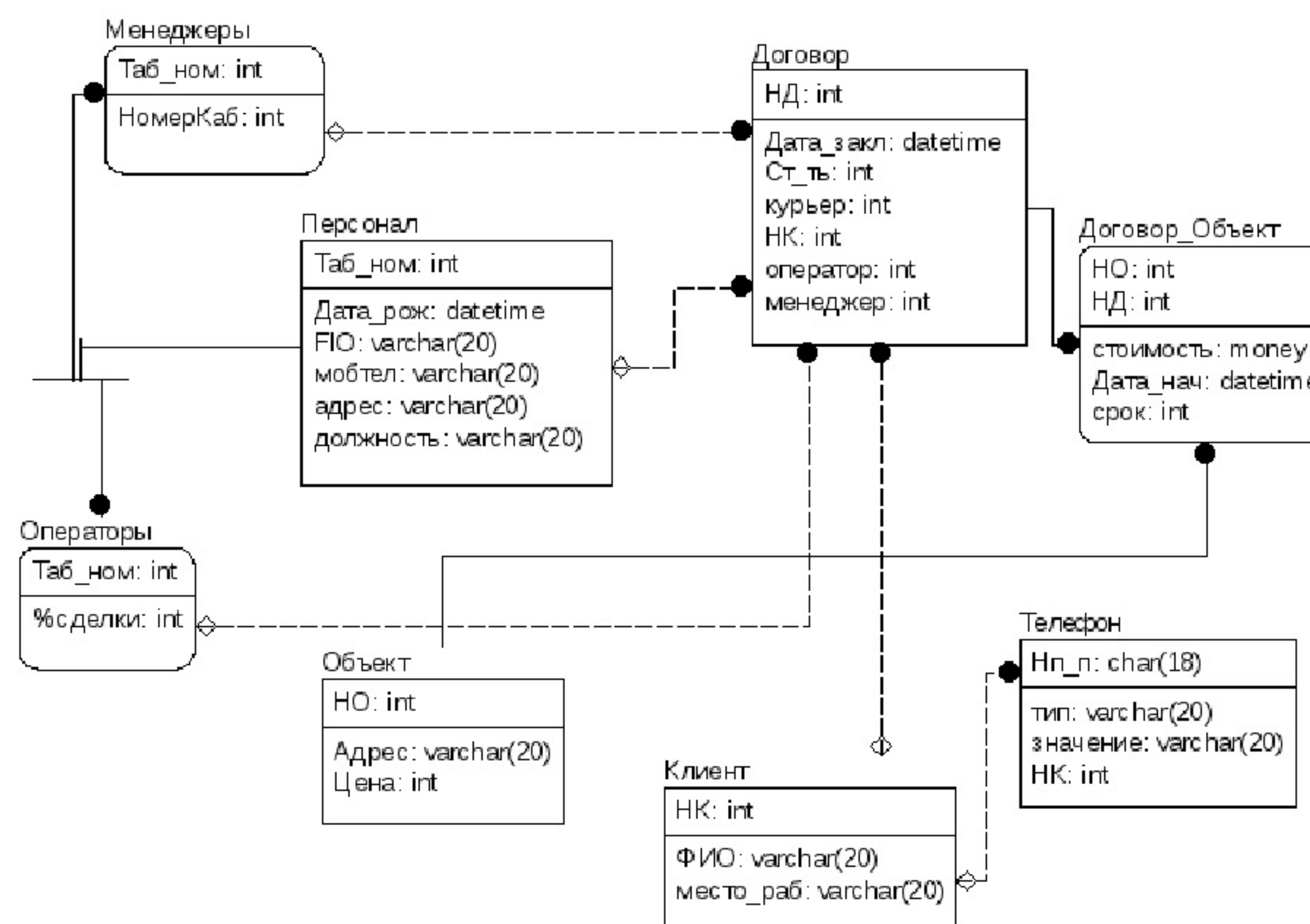
- Сущности
- Связи
- Кратности
- Атрибуты
- Ключи





**Логическая модель данных** содержит более подробную информацию, нежели концептуальная модель. На этом уровне определяются более подробные операционные и транзакционные сущности. Логическая модель не зависит от технологии, в которой она будет применяться.

- Сущности
- Связи
- Кратности
- Атрибуты
- Ключи
- Обязательность
- Уникальность
- Значение по умолчанию
- Название как в реальной БД





**Поля** — это участки таблицы, где задаются атрибуты сущностей. Под атрибутами обычно подразумеваются столбцы базы данных, которая моделируется по принципу «сущность-связь».

Bank	
	InterestRate
	LoanAmount

**Ключи** — один из способов категоризации атрибутов. Напоминаем, что ER-диаграммы помогают пользователям моделировать базы данных посредством таблиц, которые обеспечивают им упорядоченность, эффективность и высокую скорость работы. Ну а ключи применяются с целью максимально эффективно связать между собой разные таблицы в базе данных.

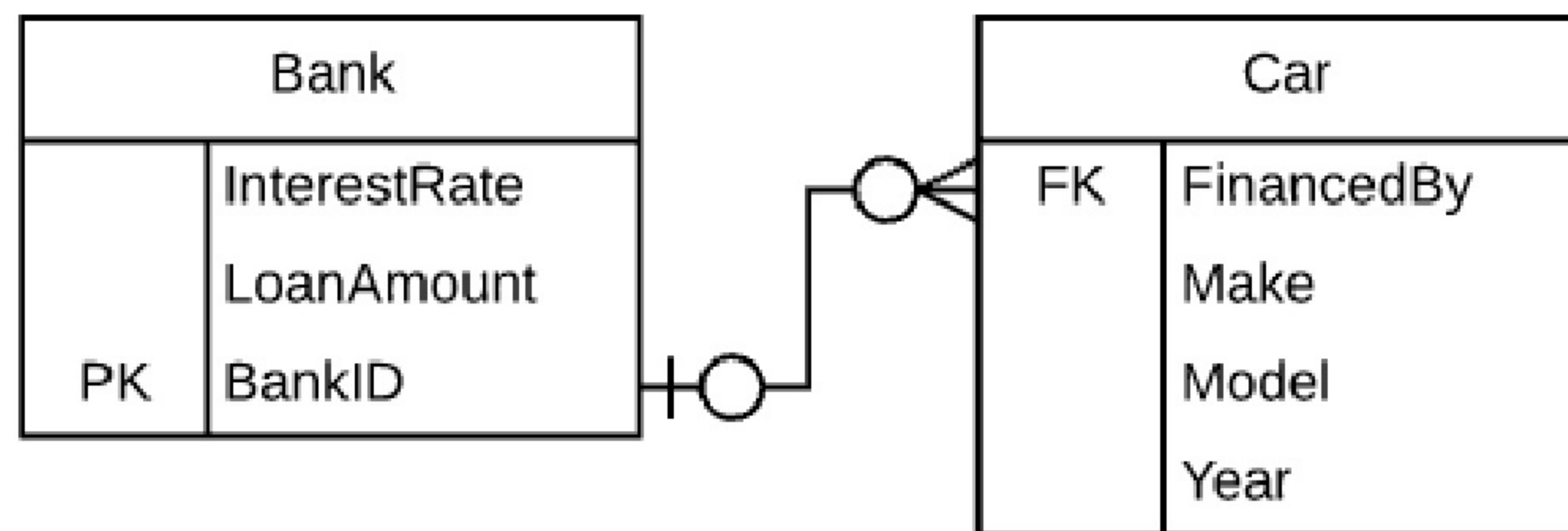


## Первичные ключи

Первичный ключ — это атрибут или сочетание атрибутов, идентифицирующих один конкретный экземпляр сущности.

## Внешние ключи

Внешний ключ создается каждый раз, когда атрибут привязывается к сущности посредством единичной или множественной связи.





One



Many



One (and only one)



Zero or one



One or many



Zero or many



## Типы

Под типом подразумевается тип данных в соответствующем поле таблицы. Однако это также может быть и тип сущности, то есть описание ее составляющих. Например, у сущности «книга» будут следующие типы: «автор», «название» и «дата публикации».

Entity
Field
Field
Field

Entity	
Key	Field
Key	Field
Key	Field

Entity	
Field	Type
Field	Type
Field	Type

Entity		
Key	Field	Type
Key	Field	Type
Key	Field	Type



**Первая нормальная форма** сводится к основным трем правилам:

- Каждая ячейка таблицы может хранить только одно значение
- Все данные в одной колонке могут быть только одного типа
- Каждая запись в таблице должна однозначно отличаться от других записей

<u>Сотрудник</u>	<u>Номер телефона</u>
Иванов И. И.	283-56-82
	390-57-34
Петров П. П.	708-62-34

<u>Сотрудник</u>	<u>Номер телефона</u>
Иванов И. И.	283-56-82
Иванов И. И.	390-57-34
Петров П. П.	708-62-34



**Вторая нормальная форма** включает в себя два требования:

- Таблица должна быть в первой нормальной форме
- Все неключевые атрибуты таблицы должны зависеть от первичного ключа

Ключевой момент второй нормальной формы - полная функциональная зависимость. Она предполагает, что атрибут В полностью функционально зависит от атрибута А, если атрибут В функционально зависит от полного значения атрибута А.

StudentId	Name	CourseId	Course	Date	TeacherId	Teacher
1	Том	1	Математика	11/06/2017	1	Смит
1	Том	2	JavaScript	14/06/2017	2	Адамс
2	Сэм	3	Алгоритмы	12/06/2017	2	Адамс
3	Боб	1	Математика	13/06/2017	1	Смит



StudentId Name

1	Том
2	Сэм
3	Боб

Таблица Courses:

CourseId	Course	TeacherId	Teacher
1	Математика	1	Смит
2	JavaScript	2	Адамс
3	Алгоритмы	2	Адамс

И таблица StudentCourses:

StudentId	CourseId	Date
1	1	11/06/2017
1	2	14/06/2017
2	3	12/06/2017
3	1	13/06/2017

Итогом стало образование связи многие ко многим (много студентов - много курсов) между таблицами Students и Courses через таблицу StudentCourses .



Требование **третьей нормальной формы** заключается в том, чтобы в таблицах отсутствовала транзитивная зависимость.

**Транзитивная зависимость** – это когда неключевые столбцы зависят от значений других неключевых столбцов.

Табельный номер	ФИО	Должность	Подразделение	Описание подразделения
1	Иванов И.И.	Программист	Отдел разработки	Разработка и сопровождение приложений и сайтов
2	Сергеев С.С.	Бухгалтер	Бухгалтерия	Ведение бухгалтерского и налогового учета финансово-хозяйственной деятельности
3	John Smith	Продавец	Отдел реализации	Организация сбыта продукции

Таблица сотрудников в третьей нормальной форме.

Табельный номер	ФИО	Должность	Подразделение
1	Иванов И.И.	Программист	1
2	Сергеев С.С.	Бухгалтер	2
3	John Smith	Продавец	3

Таблица подразделений в третьей нормальной форме.

Идентификатор подразделения	Подразделение	Описание подразделения
1	Отдел разработки	Разработка и сопровождение приложений и сайтов
2	Бухгалтерия	Ведение бухгалтерского и налогового учета финансово-хозяйственной деятельности
3	Отдел реализации	Организация сбыта продукции



<https://www.lucidchart.com/pages/ru/erd-диаграмма>

