

**Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

Отчет
по лабораторной работе №2 предмета
«Практика и Реализация Баз Данных»

Автор: Конанчук Иван Алексеевич

Факультет: ФПИн

Группа: К3240

Преподаватель:

Говорова Марина Михайловна

ИТМО

Санкт-Петербург

2025

Цель работы

Получить практические навыки анализа данных предметной области и построения инфологической модели данных базы данных методом «сущность–связь» в нотации Питера Чена–Кириллова и IDEF1X, а также описания атрибутов сущностей и ограничений целостности данных.

Практическое задание

1. Проанализировать предметную область согласно индивидуальному варианту задания.
2. Выделить сущности, их атрибуты и связи между сущностями, определить ключи и кардинальности связей.
3. Построить инфологическую модель базы данных в нотации Питера Чена–Кириллова (ER-диаграмма).
4. Реализовать полученную инфологическую модель в нотации IDEF1X в среде CA ERwin Data Modeler.
5. Составить словарь данных: дать характеристику атрибутов сущностей (типы, ключи, обязательность, ограничения целостности).
6. При наличии вычисляемых показателей описать алгоритмические связи для вычисляемых данных.

Индивидуальное задание (вариант 20)

Предметная область.

Фирмы–поставщики автомобильного топлива имеют сеть заправочных станций (АЗС и АЗГС). На автозаправках реализуется жидкое автомобильное топливо различных видов либо газ. Топливо продается по безналичному расчету с использованием специальных пластиковых карт.

База данных предназначена для анализа продаж автомобильного топлива клиентам (по видам топлива, фирмам–поставщикам, заправочным станциям), а также для анализа спроса на топливо.

Каждая фирма имеет несколько автозаправок. Каждый вид топлива поставляется несколькими фирмами–производителями. Для оплаты используется карта-счет клиента (карты имеют период действия и могут предоставлять скидку). Цены на топливо могут изменяться во времени.

Минимальный набор хранимых сведений включает:

- карту-счет клиента и сумму средств на счете;
- Ф.И.О., адрес и телефон клиента;
- код и адрес автозаправки;
- название фирмы, юридический адрес, телефон;
- код топлива, вид топлива, единицу измерения;
- цену за литр;
- дату продажи топлива и количество проданного топлива;
- код фирмы-поставщика и сроки действия цен на топливо.

Выполнение работы

Название создаваемой БД

База данных «Анализ продаж автомобильного топлива в сети АЗС»

Состав реквизитов сущностей

- Город
(id_города, название_города, код_региона)

- **Фирма**
(id_фирмы, название_фирмы, адрес_фирмы, телефон_фирмы, почта_фирмы)
- **Станция**
(id_станции, id_фирмы, id_города, тип_станции, адрес_станции, телефон_станции)
- **ТипТоплива**
(id_типа, группа, вид, марка, октановое_число, эко_класс, сезонность, стандарт)
- **ТопливоОтФирмы**
(id_топлива_фирмы, id_фирмы, id_типа, название)
- **ИсторияЦен**
(id_топлива_фирмы, дата_начала, дата_конца, цена_за_ед)
- **Клиент**
(id_клиента, фио, почта_клиента, телефон_клиента)
- **КартаСчет**
(id_карты, id_клиента, дата_выдачи, годна_до, баланс)
- **Скидка**
(id_карты, дата_начала, дата_конца, процент_скидки, тип_скидки)
- **Продажа**
(id_продажи, id_станции, id_топлива_фирмы, id_карты, дата_время, кол_во_литров, итоговая_стоимость, статус)

Схема инфологической модели в нотации Питера Чена–Кириллова

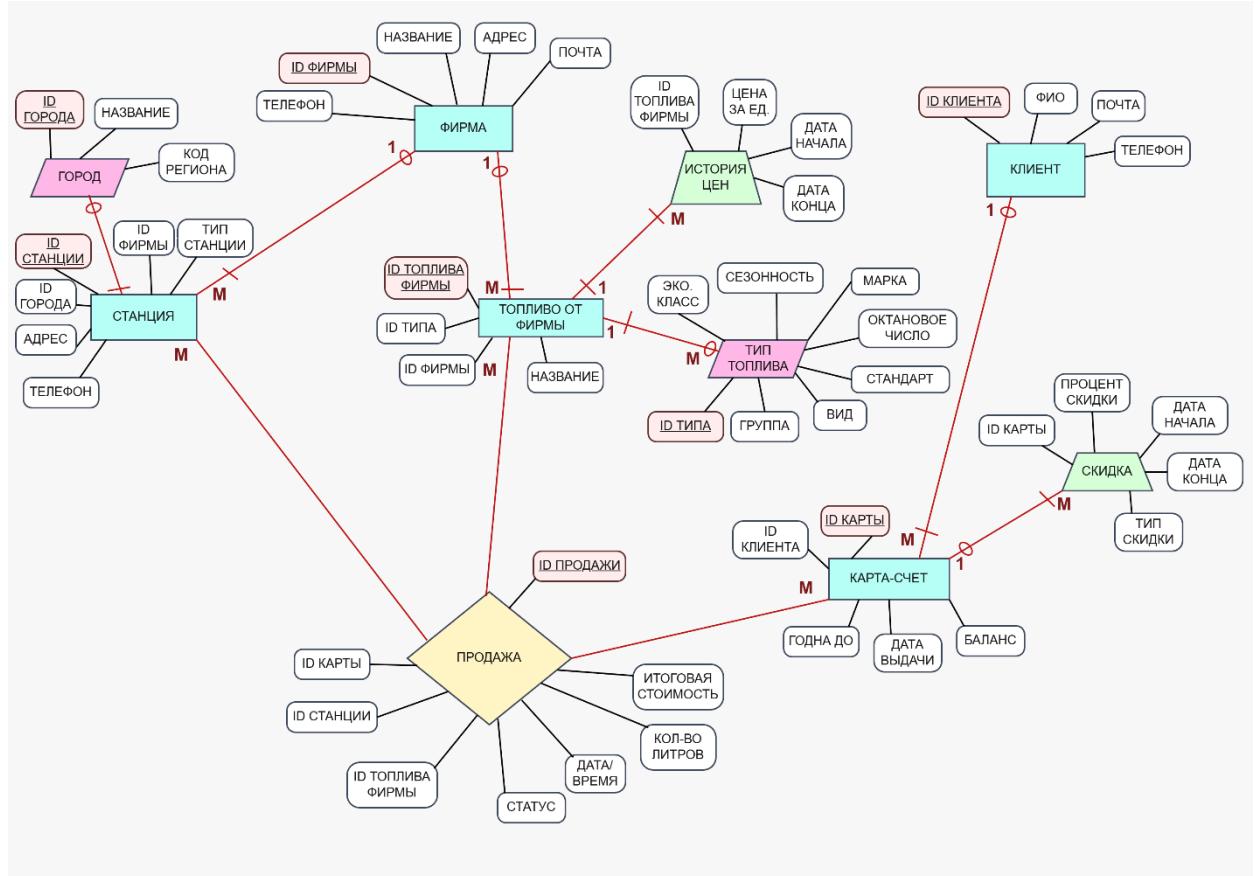
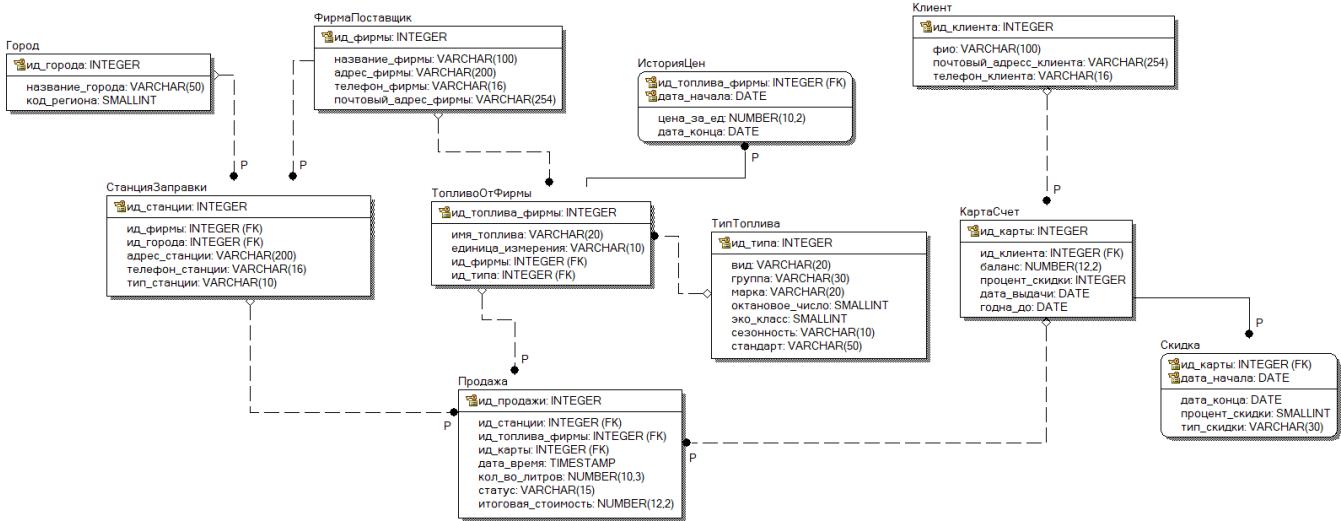


Схема инфологической модели в нотации IDEF1X

На основе ER-диаграммы в нотации Питера Чена–Кириллова выполнено моделирование в среде CA ERwin Data Modeler и построена логико-физическая модель в нотации IDEF1X.



В сущности «Продажа» хранятся количество отпущенного топлива и итоговая стоимость операции. При необходимости эти показатели также могут быть проверены/восстановлены расчетом по действующей цене из «Истории цен» и примененной скидке.

Описание атрибутов сущностей и ограничений на данные

Таблица 1:

Наименование атрибута	Тип	РК (собст в. атрибу т)	РК (внешн ий ключ)	Внешни й ключ	Обязат ельнос ть	Ограничения целостности
Сущность: Город						
id_города	INTEGER	+			+	Уникальный идентификатор города. INTEGER > 0.
название_города	VARCHAR(50)				+	Наименование города. Длина 1-50. CHECK (название_города ~ '^[A-Za-zA-Яя-Ёё0-9\s,\''()]-\Nо]+\$').
код_региона	SMALLINT				+	Код региона. CHECK (код_региона BETWEEN 1 AND 999).
Сущность: Фирма						

id_фирмы	INTEGER	+			+	Уникальный идентификатор фирмы. INTEGER > 0.
название_фирмы	VARCHAR(100)				+	Название фирмы. Длина 1-100. CHECK (название_фирмы ~ '^[A-Za-zA-ЯЁё0-9\s.,\''()]-\Nº]+\$').
адрес_фирмы	VARCHAR(200)				+	Юридический/фактический адрес. Длина 1-200. CHECK (адрес_фирмы ~ '^[A-Za-zA-ЯЁё0-9\s.,\''()]-\Nº]+\$').
телефон_фирмы	VARCHAR(16)				+	Номер телефона не целое число. Хранить в формате E.164. CHECK по регулярному выражению: ^\+[0-9]{10,15}\\$.
почта_фирмы	VARCHAR(254)				+	Электронная почта фирмы. CHECK (позиция @ > 1) и доменная часть содержит точку. Например: ^[@\s]+@[^\s]+\.[@\s]+\\$.
Сущность: Станция						
id_станции	INTEGER	+			+	Уникальный идентификатор станции. INTEGER > 0.
id_фирмы	INTEGER			+	+	FK -> Фирма(id_фирмы).

id_города	INTEGER			+	+	FK -> Город(id_города).
тип_станции	VARCHAR(10)				+	Тип станции. CHECK IN ('АЗС', 'АЗГС').
адрес_станции	VARCHAR(20 0)				+	Адрес станции. Длина 1-200. CHECK (адрес_станции ~ '^([A-Za-zA-Яа- яЁё0-9\s.,\''()]- \№]+\$').
телефон_станции	VARCHAR(16)				+	Телефон станции. Хранить в формате E.164. CHECK по регулярному выражению: ^\+?[0-9]{10,15}\$.
Сущность: Тип топлива						
id_типа	INTEGER	+			+	Уникальный идентификатор типа/марки топлива.
вид	VARCHAR(20)				+	Вид топлива. CHECK IN ('бензин', 'дизель', 'газ').
группа	VARCHAR(30)				*	Классификация (например, жидкое/газ). NULL допускается. CHECK (группа ~ '^([A-Za-zA-Яа- яЁё0-9\s.,\''()]- \№]+\$').
марка	VARCHAR(20)				+	Марка/обозначени е (например, АИ- 92, АИ-95, ДТ-3). Длина 1-20. CHECK (марка ~ '^([A-Za-zA-Яа- яЁё0-9\s.,\''()]- \№]+\$').

						яЁё0-9\s.,\"'()\-\Nº]+\$').
октановое_число	SMALLINT				*	Для бензина: CHECK (октановое_число BETWEEN 40 AND 120). Для дизеля/газа допускается NULL.
эко_класс	SMALLINT				+	Экологический класс. CHECK (эко_класс BETWEEN 0 AND 6)
сезонность	VARCHAR(10)				*	Сезонность (например, Л/Е/З/А/ВС). CHECK IN (Л','Е','З','А','ВС').
стандарт	VARCHAR(50)				*	Нормативный документ (ГОСТ/ТУ/СТО/ТР ТС). Длина до 50. CHECK (стандарт ~ '^[A-Za-zA-Яа- яЁё0-9\s.,\"'()\-\ \Nº]+\$').
Сущность: Топливо от фирмы						
id_топлива_ фирм ы	INTEGER	+			+	Уникальный идентификатор топлива конкретной фирмы. INTEGER > 0.
id_типа	INTEGER			+	+	FK -> Тип топлива(id_типа).
id_фирмы	INTEGER			+	+	FK -> Фирма(id_фирмы).

название	VARCHAR(60)				+	Коммерческое наименование (если отличается от марки). Длина 1-60. CHECK (название ~ '^[A-Zа-зА-Яа-яЁё0-9\s.,\"()]-\№]+\$').
Сущность: История цен						
id_топлива_фирмы	INTEGER		+	+	+	PK/FK -> Топливо от фирмы(id_топлива_фирмы).
дата_начала	DATE	+			+	РК. Дата начала действия цены.
дата_конца	DATE				+	Дата окончания действия цены. CHECK (дата_конца >= дата_начала).
цена_за_ед	NUMERIC(10, 2)				+	Цена за единицу (обычно литр). NUMERIC с 2 знаками после запятой. CHECK (цена_за_ед > 0).
Сущность: Клиент						
id_клиента	INTEGER	+			+	Уникальный идентификатор клиента. INTEGER > 0.
фирма	VARCHAR(100)				+	ФИО клиента. Длина 1-100. CHECK (фирма ~ '^[A-Zа-зА-Яа-яЁё\s\]-]+\$') и минимум 2 слова. (Отчество не у всех)

почта_клиента	VARCHAR(254)				+	Электронная почта клиента. CHECK (позиция @ > 1) и доменная часть содержит точку. Например: ^[@\s]+@[^\s]+\\.[^\s]+\$.
телефон_клиента	VARCHAR(16)				+	Телефон клиента. Хранить в формате E.164. CHECK по регулярному выражению: ^\+?[0-9]{10,15}\\$.
Сущность: Карта-счет						
id_карты	INTEGER	+			+	Уникальный идентификатор карты-счета. INTEGER > 0.
id_клиента	INTEGER			+	+	FK -> Клиент(id_клиента).
дата_выдачи	DATE				+	Дата выдачи карты. CHECK (дата_выдачи <= CURRENT_DATE).
годна_до	DATE				+	Дата окончания действия. CHECK (годна_до >= дата_выдачи).
баланс	NUMERIC(12,2)				+	Баланс карты. NUMERIC(12,2). CHECK (баланс >= 0).
Сущность: Скидка						
id_карты	INTEGER		+	+	+	PK/FK -> Карта-счет(id_карты).
дата_начала	DATE	+			+	PK. Дата начала действия скидки.

дата_конца	DATE				+	Дата окончания скидки. CHECK (дата_конца >= дата_начала).
процент_скидки	SMALLINT				+	Процент скидки. CHECK (процент_скидки BETWEEN 0 AND 100).
тип_скидки	VARCHAR(30)				+	Тип скидки (например, постоянная/промо/персональная). Длина до 30. CHECK (тип_скидки ~ '^[A-Za-zА-ЯяЁё0-9\\s.,\\''()]-\\№]+\$').
Ассоциативная сущность: Продажа						
id_продажи	INTEGER	+			+	Уникальный идентификатор операции продажи. INTEGER > 0.
id_карты	INTEGER			+	+	FK -> Картасчет(id_карты). Продажа по условию выполняется по безналичному расчету, поэтому NOT NULL.
id_станции	INTEGER			+	+	FK -> Станция(id_станции).
id_топлива_фирмы	INTEGER			+	+	FK -> Топливо от фирмы(id_топлива_фирмы).
дата_время	TIMESTAMP				+	Дата и время продажи с точностью до секунд. CHECK

					(дата_время <= NOW()).
кол_во_литров	NUMERIC(10, 3)			+	Количество отпущеного топлива. NUMERIC(10,3). CHECK (кол_во_литров > 0 AND кол_во_литров <= 500).
итоговая_стоимость	NUMERIC(12, 2)			+	Итоговая стоимость операции. NUMERIC(12,2). CHECK (итоговая_стоимость >= 0).
статус	VARCHAR(15)			+	Статус операции. CHECK IN ('завершена','отменена','в обработке').

Алгоритмические связи для вычисляемых данных

В модели предусмотрены показатели, которые логически присутствуют в предметной области, но могут вычисляться по данным БД и не храниться как отдельные атрибуты базовых таблиц.

1. Определение действующей цены топлива на момент продажи

Для каждой продажи необходимо определить цену топлива, действующую на дату/время продажи.

Пусть:

FuelFirm_ID(Sale) – идентификатор топлива от фирмы, связанный с продажей (Продажа.id_топлива_фирмы);

Sale_DT – дата/время продажи (Продажа.дата_время).

Тогда действующая цена определяется по таблице ИсторияЦен как:

Current_Price(Sale) = Price_Value,

где выбирается запись из таблицы ИсторияЦен, удовлетворяющая условиям:

FuelFirm_ID(PriceHistory) = FuelFirm_ID(Sale) (то есть ИсторияЦен.id_топлива_фирмы = Продажа.id_топлива_фирмы);

Date_Start <= Sale_DT;

Date_End is NULL OR Sale_DT <= Date_End.

Комментарий: если в таблице истории цен “дата конца” не задана, цена считается действующей до появления следующей записи.

2. Вычисление стоимости продажи

Пусть для каждой продажи задается количество реализованного топлива:

Fuel_Amount(Sale) – количество литров (Продажа.кол_во_литров).

Тогда стоимость продажи без учета скидки:

Sale_Sum(Sale) = Fuel_Amount(Sale) * Current_Price(Sale).

В модели допускается как хранение **Sale_Sum** (например Продажа.итоговая_стоимость), так и вычисление в запросах/отчетах. Если хранится – должно соответствовать расчету на момент фиксации продажи.

3. Применение скидки по карте-счету

Пусть:

Card_ID(Sale) – карта, участвующая в продаже (Продажа.id_карты);

Sale_DT – дата/время продажи;

Card_Discount_Prc(Sale) – процент скидки, действующий для карты на момент продажи.

Тогда **Card_Discount_Prc(Sale)** определяется по таблице Скидка как процент скидки из записи, удовлетворяющей:

Скидка.id_карты = Продажа.id_карты;

Скидка.дата_начала <= Sale_DT;

Скидка.дата_конца is NULL OR Sale_DT <= Скидка.дата_конца.

Далее:

Discount_Value(Sale) = Sale_Sum(Sale) * Card_Discount_Prc(Sale) / 100;

Sale_Sum_With_Discount(Sale) = Sale_Sum(Sale) - Discount_Value(Sale).

Комментарий: если подходящей скидки нет, то **Card_Discount_Prc(Sale) = 0.**

4. Изменение баланса карты-счета

Пусть:

Old_Balance(Card) – баланс до покупки;

New_Balance(Card) – баланс после покупки.

Тогда:

New_Balance(Card) = Old_Balance(Card) - Sale_Sum_With_Discount(Sale).

В БД можно:

либо хранить только факт продажи, а баланс пересчитывать как агрегат по всем операциям (строже, но тяжелее),

либо хранить баланс в Карта-счет и обновлять его при фиксации продажи (быстрее для чтения, но требует контроля целостности).

5. Определение локального времени сделки по часовому поясу города

В актуальной модели у сущности Город хранится код_региона, однако по одному коду региона невозможно однозначно получить смещение UTC (в России один регион может иметь разные часовые пояса, и наоборот). Поэтому для вычисления локального времени необходима таблица соответствия или справочник часовых поясов.

Пусть:

City_ID(Station) – город станции (Станция.id_города);

Region_Code(City) – код региона (Город.код_региона);

TZ_Map(Region_Code) → TZ_Offset – таблица/функция соответствия кода региона и смещения UTC (или, лучше, IANA timezone);

Sale_DateTime_UTC – момент продажи в UTC;

Sale_DateTime_Local – локальное время.

Тогда:

Sale_DateTime_Local = Sale_DateTime_UTC + TZ_Offset(Region_Code(City)).

Комментарий: для корректного учета сезонных переводов времени предпочтительнее хранить не **TZ_Offset**, а **TimeZone_Name** (например Europe/Moscow) и получать локальное время средствами СУБД.

Выводы

В ходе работы выполнен анализ предметной области, связанной с продажей автомобильного топлива на сети АЗС фирмы-поставщика. Выделены основные сущности (фирма, город, станция заправки, клиент, карта-счет, тип топлива, история цен, продажа, топливо от фирмы, скидка), их атрибуты и связи между ними.

На основе анализа построена инфологическая модель базы данных в нотации Питера Чена–Кириллова, отражающая сущности, связи и кардинальности. Затем модель реализована в нотации IDEF1X в среде CA ERwin Data Modeler с указанием ключей, типов данных и ограничений целостности.

Сформирован словарь данных (таблица 1), описывающий типы и ограничения атрибутов. Определены алгоритмические связи для вычисляемых показателей, связанных с расчетом действующей цены топлива, стоимости продажи, скидок по карте-счету и изменением баланса карты.