**Лабораторная работа №3. Формы бухгалтерского учета (Т2,Т4,Т9,Т12) . Задача преобразовать форму в набор таблиц сущностей.**

Питер Чен. Четыре уровня представления моделей данных:

1. Информация об объектах и связях предметной области (концептуальная модель)
2. Структурированная информация о предметной области (логическая модель)
3. Структуры данных, не зависящие от способа доступа (физическая модель)
4. Структуры данных, зависящие от способа доступа (модель аппаратного уровня)

Перед созданием базы данных разработчик должен определить, из каких таблиц должна состоять база данных, какие данные нужно поместить в каждую таблицу, как связать таблицы. Эти вопросы решаются на этапе проектирования базы данных. В результате проектирования должна быть определена логическая структура базы данных, то есть состав реляционных таблиц, их структура и межтабличные связи. Перед созданием базы данных необходимо располагать описанием выбранной предметной области, которое должно охватывать реальные объекты и процессы, определить все необходимые источники информации для удовлетворения предполагаемых запросов пользователей и определить потребности в обработке данных. На основе такого описания на этапе проектирования базы данных определяются состав и структура данных предметной области, которые должны находиться в БД и обеспечивать выполнение необходимых запросов и задач пользователей. Структура данных предметной области может отображаться информационно-логической моделью. На основе этой модели легко создается реляционная база данных

Логический уровень означает прямое отображение фактов из реальной жизни. Например, люди, столы, отделы и компьютеры являются реальными объектами. Они именуются на естественном языке, с любыми разделителями слов (пробелы, запятые и т.д.). На логическом уровне не рассматривается использование конкретной СУБД, не определяются типы данных (например, целое или вещественное число) и не определяются индексы для таблиц.

Создание физической модели

Целевая СУБД, имена объектов и типы данных, индексы составляют второй (физический) уровень модели ERwin. ERwin предоставляет возможности создавать и управлять этими двумя различными уровнями представления одной диаграммы (модели), равно как и иметь много вариантов отображения на каждом уровне

Целью создания физической модели является обеспечение администратора соответствующей информацией для переноса логической модели данных в СУБД. При этом логическая модель трансформируется в физическую по следующему принципу: сущности становятся таблицами, атрибуты становятся столбцами, а ключи становятся индексами (см. табл.1).

Таблица 1 - Сопоставление компонентов логической и физической модели

|  |  |
| --- | --- |
| Логическая модель | Физическая модель |
| Сущность | Таблица |
| Атрибут | Столбец |
| Логический тип  (текст, число, дата, blob) | Физический тип (корректный тип, зависящий от выбранной СУБД) |
| Первичный ключ | Первичный ключ, индекс PK |
| Внешний ключ | Внешний ключ, индекс FK |
| Альтернативный ключ | AK-индекс, уникальный,  непервичный индекс |
| Правило бизнес-логики | Триггер или сохраненная  процедура |
| Взаимосвязи | Взаимосвязи, определяемые использованием FK-атрибутов |

• Сущности, атрибуты, группы ключей и домены можно создавать только на логическом уровне модели.

• Таблицы, столбцы, индексы и домены можно создавать только на физическом уровне.

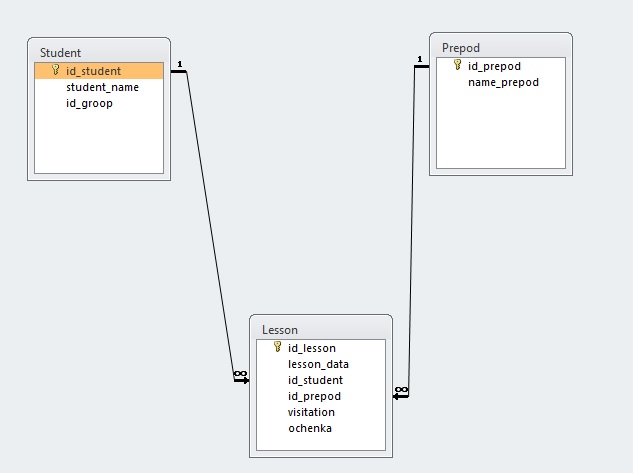
**Задание**

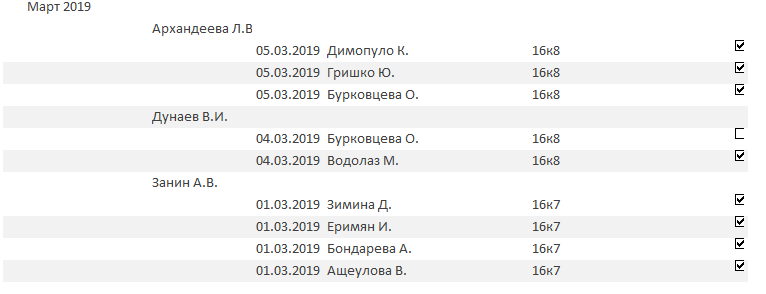
1. Построить упрощенную логическую модель для работника предприятия, оперирующего следующими формами бухгалтерского учета (Т2, Т4, T9, T12, др.).
2. Преобразовать формы в набор таблиц/сущностей, оформить связи, построить физическую модель, установить связи и типы данных (ERWin, Access).

Пример: представление отчётной формы «Индивидуальный журнал преподавателя» в виде наборов таблиц.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Факультет |  | Дисциплина |  | Группа |  |
| Кафедра |  | Курс |  | Преподаватель |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ф.И.О.  студента | Занятие | | | Занятие | | | Занятие | | |
| Содержание | | | Содержание | | | Содержание | | |
| Дата | | | Дата | | | Дата | | |
| посещ. |  | выполн. | посещ. |  | выполн. | посещ. |  | выполн. |
|  |  |  |  |  |  |  |  |  |  |

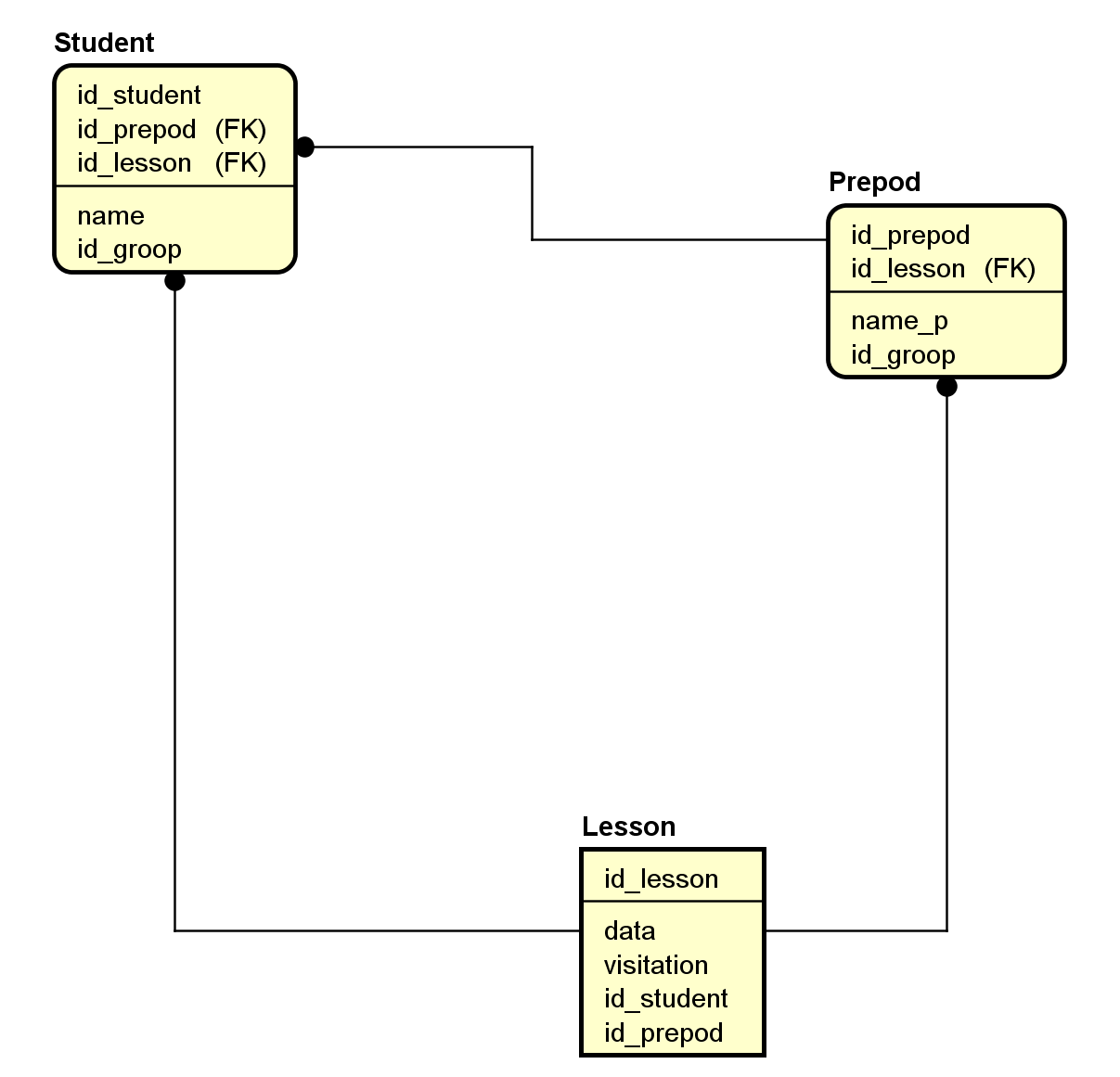




Инструмент проектирования системы - **Astah** **Professional**

*Astah UML is* a modeling tool that supports UML (Unified Modeling Language) 2.x (partly), UML1.4 and Mind Map.

!!! УСТРАНИТЕ НЕКОРРЕКТНЫЕ СВЯЗИ И СООТВЕТСТВУЮЩИЕ ИМ КОМАНДЫ СКРИПТА.



CREATE TABLE Lesson (

id\_lesson CHAR(10) NOT NULL,

data CHAR(10),

visitation CHAR(10),

id\_student CHAR(10),

id\_prepod CHAR(10)

);

ALTER TABLE Lesson ADD CONSTRAINT PK\_Lesson PRIMARY KEY (id\_lesson);

CREATE TABLE Prepod (

id\_prepod CHAR(10) NOT NULL,

id\_lesson CHAR(10) NOT NULL,

name\_p CHAR(10),

id\_groop CHAR(10)

);

ALTER TABLE Prepod ADD CONSTRAINT PK\_Prepod PRIMARY KEY (id\_prepod,id\_lesson);

CREATE TABLE Student (

id\_student CHAR(10) NOT NULL,

id\_prepod CHAR(10) NOT NULL,

id\_lesson CHAR(10) NOT NULL,

name CHAR(10),

id\_groop CHAR(10)

);

ALTER TABLE Student ADD CONSTRAINT PK\_Student PRIMARY KEY (id\_student,id\_prepod,id\_lesson);

ALTER TABLE Prepod ADD CONSTRAINT FK\_Prepod\_0 FOREIGN KEY (id\_lesson) REFERENCES Lesson (id\_lesson);

ALTER TABLE Student ADD CONSTRAINT FK\_Student\_0 FOREIGN KEY (id\_prepod,id\_lesson) REFERENCES Prepod (id\_prepod,id\_lesson);

ALTER TABLE Student ADD CONSTRAINT FK\_Student\_1 FOREIGN KEY (id\_lesson) REFERENCES Lesson (id\_lesson);

Инструмент проектирования системы - **Astah** **Professional**

*Astah UML is* a modeling tool that supports UML (Unified Modeling Language) 2.x (partly), UML1.4 and Mind Map.

|  |
| --- |
| **ПРИМЕР – БД ФАКУЛЬТАТИВОВ ДЛЯ СТУДЕНТОВ**  Концептуальная модель, база данных хранит информацию о:  1. Всех факультативах  2. Студентах, посещающих факультативы  3. Преподавателях факультативов  4. Списке тем для каждого факультатива  Сущность "Факультатив" содержит поля:  -название факультатива  -время проведения  -ФИО преподаввателя  -количество мест  Сущность "Преподаватель" содержит поля:  -ФИО преподавателя  Сущность "Студент" содержит поля:  -ФИО студента  -номер группы  Сущность "Список тем" содержит поля:  -название темы  -количество часов, отведённых на тему    CREATE TABLE Elective (  id\_elective CHAR(10) NOT NULL,  name\_el CHAR(10),  time\_el CHAR(10),  prepod CHAR(10),  num\_of\_seats CHAR(10)  );  ALTER TABLE Elective ADD CONSTRAINT PK\_Elective PRIMARY KEY (id\_elective);  CREATE TABLE List\_of\_themes (  id\_theme CHAR(10) NOT NULL,  id\_elective CHAR(10) NOT NULL,  name\_th CHAR(10),  number\_of\_classes CHAR(10)  );  ALTER TABLE List\_of\_themes ADD CONSTRAINT PK\_List\_of\_themes PRIMARY KEY (id\_theme,id\_elective);  CREATE TABLE Prepod (  id\_prepod CHAR(10) NOT NULL,  id\_elective CHAR(10) NOT NULL,  name\_pr CHAR(10)  );  ALTER TABLE Prepod ADD CONSTRAINT PK\_Prepod PRIMARY KEY (id\_prepod,id\_elective);  CREATE TABLE Student (  id\_student CHAR(10) NOT NULL,  id\_elective CHAR(10) NOT NULL,  name\_st CHAR(10),  group CHAR(10)  );  ALTER TABLE Student ADD CONSTRAINT PK\_Student PRIMARY KEY (id\_student,id\_elective);  ALTER TABLE List\_of\_themes ADD CONSTRAINT FK\_List\_of\_themes\_0 FOREIGN KEY (id\_elective) REFERENCES Elective (id\_elective);  ALTER TABLE Prepod ADD CONSTRAINT FK\_Prepod\_0 FOREIGN KEY (id\_elective) REFERENCES Elective (id\_elective);  ALTER TABLE Student ADD CONSTRAINT FK\_Student\_0 FOREIGN KEY (id\_elective) REFERENCES Elective (id\_elective);  **ПРИМЕР – БД ПРОКАТ АВТОМОБИЛЕЙ**    CREATE TABLE Автомобильный салон (  код\_салона CHAR(10) NOT NULL,  Город CHAR(10),  Улица CHAR(10)  );  ALTER TABLE Автомобильный салон ADD CONSTRAINT PK\_Автомобильный салон PRIMARY KEY (код\_салона);  CREATE TABLE Арендатор (  код\_арендатора CHAR(10) NOT NULL,  Имя CHAR(10),  Фамилия CHAR(10),  Дата рождения CHAR(10),  Адрес CHAR(10),  Телефон CHAR(10)  );  ALTER TABLE Арендатор ADD CONSTRAINT PK\_Арендатор PRIMARY KEY (код\_арендатора);  CREATE TABLE Автомобиль (  код\_авто CHAR(10) NOT NULL,  код\_салона CHAR(10) NOT NULL,  Марка CHAR(10),  Страна сборки CHAR(10),  Цвет CHAR(10)  );  ALTER TABLE Автомобиль ADD CONSTRAINT PK\_Автомобиль PRIMARY KEY (код\_авто,код\_салона);  CREATE TABLE Авто в прокате (  код\_взятого\_авто CHAR(10) NOT NULL,  код\_арендатора CHAR(10) NOT NULL,  код\_авто CHAR(10) NOT NULL,  код\_салона CHAR(10) NOT NULL,  Дата взятия CHAR(10),  Вернуть до CHAR(10),  Возвращен CHAR(10),  Штраф CHAR(10)  );  ALTER TABLE Авто в прокате ADD CONSTRAINT PK\_Авто в прокате PRIMARY KEY (код\_взятого\_авто,код\_арендатора,код\_авто,код\_салона);  ALTER TABLE Автомобиль ADD CONSTRAINT FK\_Автомобиль\_0 FOREIGN KEY (код\_салона) REFERENCES Автомобильный салон (код\_салона);  ALTER TABLE Авто в прокате ADD CONSTRAINT FK\_Авто в прокате\_0 FOREIGN KEY (код\_арендатора) REFERENCES Арендатор (код\_арендатора);  ALTER TABLE Авто в прокате ADD CONSTRAINT FK\_Авто в прокате\_1 FOREIGN KEY (код\_авто,код\_салона) REFERENCES Автомобиль (код\_авто,код\_салона);  !!! НЕ ЗАБУДЬТЕ УДАЛИТЬ СОЗДАННЫЕ ТЕСТОВЫЕ ТАБЛИЦЫ И ПОСЛЕДОВАТЕЛЬНОСТИ !!! |

# Описание схемы HR

## Оригинальное описание из документации Oracle

Ниже приводится описание, данное в документе Oracle® Database. Sample Schemas. 10g Release 2 (10.2). B14198-01. June 2005

In the Human Resource (HR) records, each employee has an identification number, e-mail address, job identification code, salary, and manager. Some employees earn commissions in addition to their salary.

The company also tracks information about jobs within the organization. Each job has an identification code, job title, and a minimum and maximum salary range for the job.

Some employees have been with the company for a long time and have held different positions within the company. When an employee resigns, the duration the employee was working, the job identification number, and the department are recorded.

The sample company is regionally diverse, so it tracks the locations of its warehouses and departments. Each employee is assigned to a department, and each department is identified either by a unique department number ~~or a short name~~[[1]](#footnote-1). Each department is associated with one location, and each location has a full address that includes the street name, postal code, city, state or province, and the country code.

In places where the departments and warehouses are located, the company records details such as, the country name, ~~currency symbol, currency name~~[[2]](#footnote-2), and the region where the country resides geographically.

## Перевод оригинального описания из документации Oracle

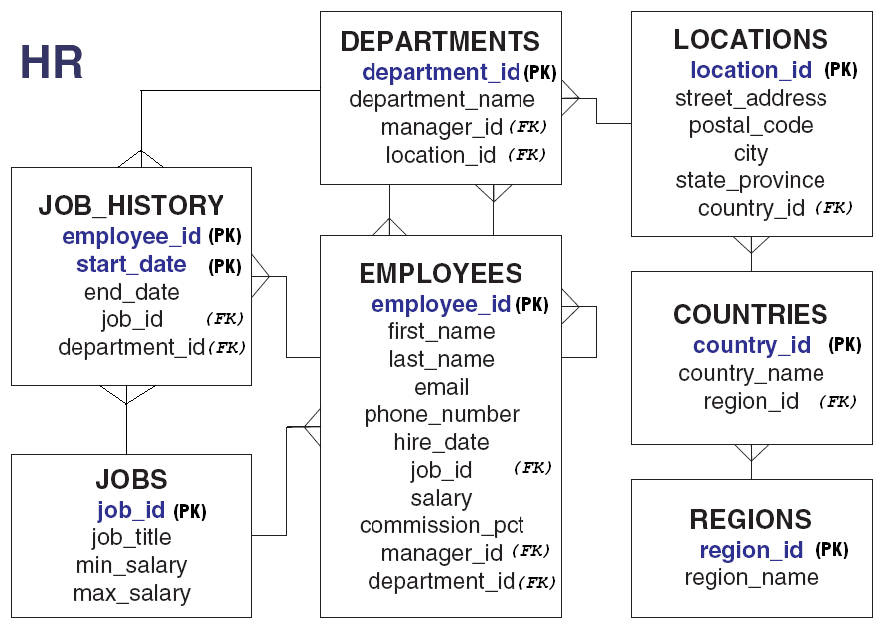
В записях о штате сотрудников каждый сотрудник имеет идентификационный номер, адрес электронной почты, идентификатор должности, оклад и (идентификатор) руководителя. Некоторые сотрудники дополнительно к окладу зарабатывают комиссионные.

Также компания хранит информацию о должностях в рамках организации. У каждой должности есть идентификатор, название и диапазон ("вилка") окладов.

Некоторые сотрудники долгое время работают в компании и занимали в ней разные должности. При уходе сотрудника с должности записываются сведения о продолжительности его работы в данной должности, её идентификатор и подразделение, в котором работал сотрудник.

Компания из данного примера работает в нескольких регионах, поэтому хранятся сведения о местах расположения её складов и подразделений. Каждый сотрудник приписан к одному подразделению, а каждое подразделение идентифицируется уникальным номером ~~или кратким названием~~[[3]](#footnote-3). Каждое подразделение связано (находится во взаимно однозначном соответствии) с одним местоположением, а для каждого местоположения хранится полный адрес, содержащий название улицы, почтовый индекс, город, штат или область и код страны.

Для местоположений подразделений и складов хранится детализированная информация: название страны, ~~символ валюты, название валюты~~[[4]](#footnote-4) и географический регион, где расположена страна.



Prompt \*\*\*\*\*\* Creating REGIONS table ....

CREATE TABLE regions

( region\_id NUMBER

CONSTRAINT region\_id\_nn NOT NULL

, region\_name VARCHAR2(25)

);

CREATE UNIQUE INDEX reg\_id\_pk

ON regions (region\_id);

ALTER TABLE regions

ADD ( CONSTRAINT reg\_id\_pk

PRIMARY KEY (region\_id)

) ;

Prompt \*\*\*\*\*\* Creating COUNTRIES table ....

CREATE TABLE countries

( country\_id CHAR(2)

CONSTRAINT country\_id\_nn NOT NULL

, country\_name VARCHAR2(40)

, region\_id NUMBER

, CONSTRAINT country\_c\_id\_pk

PRIMARY KEY (country\_id)

)

ORGANIZATION INDEX;

ALTER TABLE countries

ADD ( CONSTRAINT countr\_reg\_fk

FOREIGN KEY (region\_id)

REFERENCES regions(region\_id)

) ;

Prompt \*\*\*\*\*\* Populating REGIONS table ....

INSERT INTO regions VALUES

( 1

, 'Europe'

);

INSERT INTO regions VALUES

( 2

, 'Americas'

);

INSERT INTO regions VALUES

( 3

, 'Asia'

);

INSERT INTO regions VALUES

( 4

, 'Middle East and Africa'

);

Prompt \*\*\*\*\*\* Populating COUNTIRES table ....

INSERT INTO countries VALUES

( 'IT'

, 'Italy'

, 1

);

INSERT INTO countries VALUES

( 'JP'

, 'Japan'

, 3

);

INSERT INTO countries VALUES

( 'US'

, 'United States of America'

, 2

);

INSERT INTO countries VALUES

( 'CA'

, 'Canada'

, 2

);

INSERT INTO countries VALUES

( 'CN'

, 'China'

, 3

);

INSERT INTO countries VALUES

( 'IN'

, 'India'

, 3

);

INSERT INTO countries VALUES

( 'AU'

, 'Australia'

, 3

);

1. Неверно. Нет соответствующего ограничения целостности [↑](#footnote-ref-1)
2. Неверно. Ни в одной из таблиц схемы HR нет столбцов, где можно хранить данную информацию [↑](#footnote-ref-2)
3. Неверно. Нет соответствующего ограничения целостности [↑](#footnote-ref-3)
4. Неверно. Ни в одной из таблиц схемы HR нет столбцов, где можно хранить данную информацию [↑](#footnote-ref-4)