ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ.ПРОФ.М.А.БОНЧ-БРУЕВИЧА»

(СПбГУТ)

**Кафедра безопасности информационных систем**

**Пояснительная записка к курсовому проекту**

**на тему**

**«Информационная система школы иностранных языков»**

по дисциплине «Управление данными»

Выполнил:

« » г. \_\_\_\_\_\_\_\_\_\_\_/

Принял:

« » г. \_\_\_\_\_\_\_\_\_\_\_/

**Санкт-Петербург  
2020**

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc514838533)

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc514838534)

[2 ИНФОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ 8](#_Toc514838535)

[2.1 Инфологическая модель 8](#_Toc514838536)

[2.2 Нормализация отношений 12](#_Toc514838537)

[3 ДАТОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ 13](#_Toc514838538)

[3.1 Физическая модель 13](#_Toc514838539)

[4 ПРОВЕРКА НА ОШИБКИ 15](#_Toc514838540)

[5 ПРЯМОЕ ПРОЕКТИРОВАНИЕ 16](#_Toc514838541)

[6 ЗАПОЛНЕНИЕ ТАБЛИЦ 17](#_Toc514838542)

[6.1 Заполнение таблиц данными 17](#_Toc514838543)

[6.2 Работоспособность ограничений 20](#_Toc514838544)

[7 ВЫПОЛНЕНИЕ SQL – ЗАПРОСОВ 21](#_Toc514838545)

[ЗАКЛЮЧЕНИЕ 25](#_Toc514838546)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 26](#_Toc514838547)

## ВВЕДЕНИЕ

Динамика. Именно этим словом можно охарактеризовать современный мир. Большое значение в нем играет информация. Ее объемы увеличиваются, а методы ее обработки изменяются. В связи с большим распространением информации в нашу жизнь стали внедрятся информационные технологии. Они стали частью нашего мира. Мы сталкиваемся с ними ежедневно. Поездка на метро, поход в супермаркет или ресторан, использование интернета и даже оплата государственных налогов – все эти действия не обойдутся без информационных систем. Записываясь на прием к врачу, оформляя банковскую карту или просто совершая покупку в интернет магазине, мы оставляем о себе большое количество важной информации, которую необходимо хранить, защищать, а иногда изменять и передавать. В обработке и хранении информации может быть заинтересован как один человек, так и целая организация.

Актуальность курсовой работы заключается в том, что гостиничный комплекс является примером организации, заинтересованной в обработке и хранении информации. Почти каждый человек сталкивался с оказанием услуг в этой области, ведь современные люди очень часто путешествуют. Время пребывания в незнакомом городе по своим личным или служебным делам всегда наполнено множеством интересных и важных событий. Чтобы отдохнуть и набраться сил для нового дня, люди останавливаются в гостиничных номерах. Для качественного и быстрого обслуживания всех клиентов гостиничный комплекс использует базы данных. Благодаря ним можно хранить большое количество информации о клиентах, ее конструировать и изменять. Также именно использование базы данных поможет вести некий учет. Получать нужную информацию, например, о количестве свободных или занятых номеров, числе постояльцев, жалобах, стоимости конкретного номера и т.д.

Цель курсовой работы: разработка информационной системы гостиничного комплекса. Задачи: описание предметной области, создание логической и физической моделей в Erwin Data Modeler, проектирование базы данных в Oracle SQL Developer.

Использование такой информационной системы значительно облегчает работу администратора гостиничного комплекса.

1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Предметная область – область применения конкретной базы данных.

Предметная область определена, если известны существующие в ней объекты, их свойства и отношения.

Объект – это то, о чем должна накапливаться информация.

При определении границ предметной области используют подход «от реального мира», с помощью экспертов и подход «от запросов пользователей».

Для определения предметной области в курсовой работе использовались оба подхода.

В данной работе предметной областью является школа изучения иностранных языков. База данных разработана для школы, и она значительно упростит работу администратора.

Основополагающей сущностью является человек, который характеризуется именем, фамилией, и номером паспорта. Человек может исполнять роль рабочего и роль студента, при этом один человек может занимать разные должности вне зависимости от филиала.

Рабочий представляет из себя человека, который занимает определенную должность, работает в определенном филиале, и получает зарплату. Так же при приёме на работу с человеком заключается договор, номер которого является характеристикой рабочего.

Должность представляет из себя текстовое описание профессии.

Студент представляет из себя человека, который обучается в определенной учебной группе.

Учебная группа представляет из себя группу студентов, посещающих занятия. Учебная группа может состоять из одного человека. Характеризуется тем, какой язык изучается в этой группе, в каком году группа была сформирована, а также тем, какой преподаватель проводит занятия.

Филиалы школы расположены в одном городе и каждый филиал характеризуется названием улицы, на котором расположен филиал, номером дома, контактным телефоном, а также временем открытия филиала и, в случае если филиал был расформирован, временем закрытия филиала.

Студенты в учебных группах и преподаватели посещают занятия. Занятия описываются номером группы, у которой проходит занятия, а также временем начала занятия и аудиторией. Предполагается, что каждое занятие в школе длится один академический час.

1. ИНФОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ

2.1 Инфологическая модель

Цель инфологического моделирования – создание структурированной инфологической модели предметной области.

Инфологическая модель применяется па втором этапе проектирования БД, после словесного описания предметной области.

Основные элементы инфологической модели: сущности, атрибуты и связи.

Сущность – это некоторая абстракция реально существующего объекта, предмета, явления.

Атрибут – поименованная характеристика сущности.

Связь – средство представления отношений между сущностями, ассоциирование двух или более сущностей.

Определим сущности и их атрибуты:

1. Сущность “Филиал” – BRANCH. Атрибуты:
   1. BRANCH\_ID – Уникальный идентификатор – первичный ключ;
   2. STREET – название улицы, на которой расположен филиал;
   3. BUILDING – номер здания, в котором расположен филиал;
   4. PHONE – контактный телефон;
   5. CREATED – дата открытия филиала;
   6. UNFORMED – дата расформирования филиала, если филиал закрыт.
2. Сущность “Человек” – PERSON. Атрибуты:
   1. PERSON\_ID – Уникальны идентификатор – первичный ключ;
   2. PASSPORT – номер и серия паспорта;
   3. NAME – имя;
   4. SURNAME – фамилия.
3. Сущность “Должность” – POSITION. Атрибуты:
   1. POSITION\_ID – Уникальный идентификатор – первичный ключ;
   2. DESCRIPTION – Описание должности.
4. Сущность “Работник” – EMPLOYEE. Атрибуты:
   1. EMPLOYEE\_ID – Уникальный идентификатор – первичный ключ;
   2. PERSON\_ID – Идентификатор личности, внешний ключ;
   3. Contract\_id – Номер договора;
   4. BRANCH\_ID – Идентификатор филиала, внешний ключ;
   5. Position\_id – Идентификатор должности, внешний ключ;
   6. SALARY – Зарплата.
5. Сущность “Группа” – COURSE. Атрибуты:
   1. COURSE\_ID – Уникальный идентификатор – первичный ключ;
   2. LANGUAGE – Изучаемый язык;
   3. EMPLOYEE\_ID – Идентификатор преподавателя, внешний ключ;
   4. YEAR\_SINCE – Год, в котором была сформирована группа.
6. Сущность “Студент” – STUDENT. Атрибуты:
   1. STUDENT\_ID – Уникальный идентификатор – первичный ключ;
   2. PERSON\_ID – Идентификатор личности, внешний ключ;
   3. COURSE\_ID – Идентификатор группы, внешний ключ.
7. Сущность “Занятие” – LESSON. Атрибуты:
   1. LESSON\_ID – Уникальный идентификатор – первичный ключ;
   2. COURSE\_ID – Идентификатор группы, внешний ключ.
   3. STARTING – Дата и время начала занятия;
   4. ROOM – Номер аудитории.

Определим связи между сущностями:

Один работник представляет из себя одного человека. Так же при принятии одного человека на новую специальность для одного человека создается новая сущность “работник”, так как подписывается новый контракт и человек начинает выполнять другие обязанности. Следовательно, между сущностями “работник” и “человек” связь многие-к-одному, между сущностями “должность” и “работник” связь один-ко-многим. Между сущностями “работник” – “филиал” установлена связь многие-к-одному.

У одного преподавателя может заниматься несколько групп, следовательно, между сущностями “группа” и “преподаватель” связь многие-к-одному.

Каждый студент представляет собой одного человека и в одной группе может находиться несколько человек, следовательно, связь между сущностями “студент” и “человек” – один-к-одному, а между “студент” и “группа” – многие-к-одному.

Занятия предполагают присутствие одной учебной группы, следовательно, между сущностями “занятие” и “группа” связь один-к-одному.

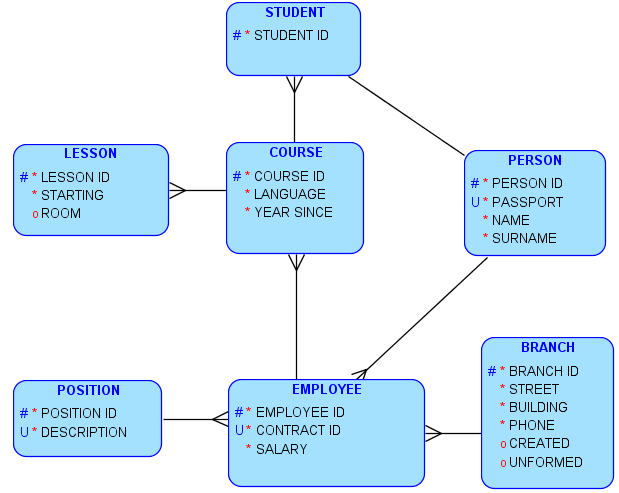


Рисунок 1. Логическая модель в системе sqldeveloper.

## Нормализация отношений

Главная цель нормализации отношений – устранение избыточности и дублирования информации.

Нормализация – процесс последовательного разбиения и преобразования исходного набора таблиц для построения набора взаимосвязанных таблиц в нормальных формах.

Отношение находится в 1НФ, если все значения его атрибутов атомарны;

Отношение находится в 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут функционально полно зависит от ключа (простого или составного);

Отношение находится в 3НФ, если оно находится во 2НФ и в нем отсутствуют транзитивные зависимости не ключевых атрибутов от ключа.

В разработанной модели все отношения находятся в 1 НФ, так как все значения атомарны.

Все отношения содержат простые первичные ключи, поэтому они находятся во 2 НФ.

Во всех отношениях отсутствуют транзитивные зависимости, поэтому отношения находятся в 3 НФ.

1. ДАТЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

## Физическая модель и прямое проектирование

**На основании построенной логической модели была спроектирована физическая модель. DDL sql-запросы представлены в приложении А.** На рисунке 2 представлена ER-диаграмма разработанной модели.

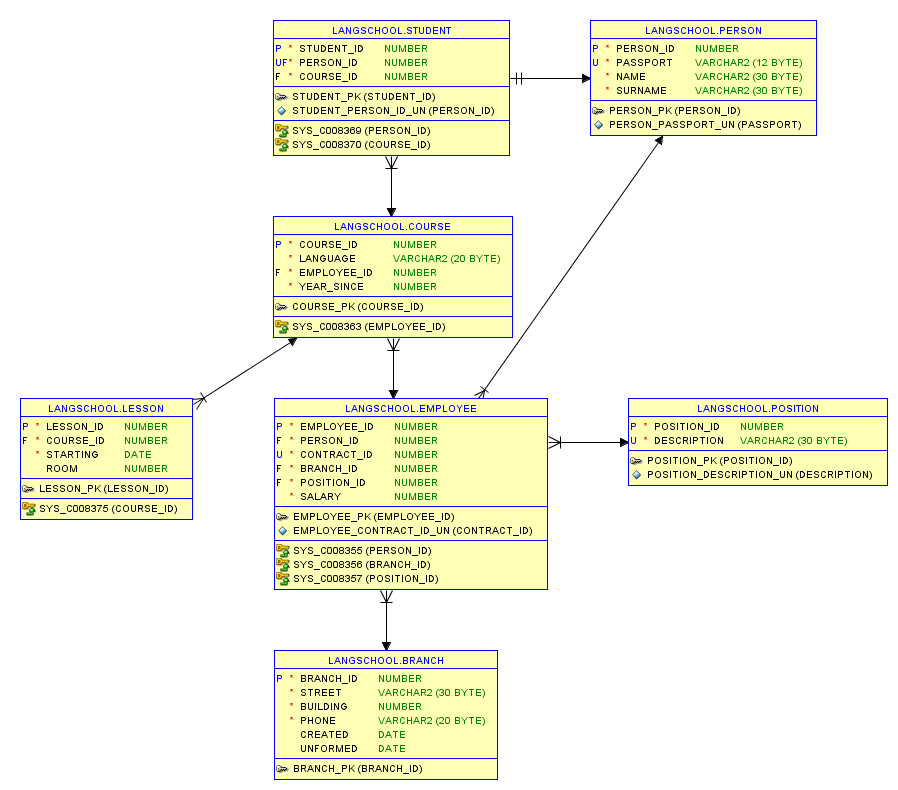


Рисунок 2. ER-диаграмма

1. ЗАПОЛНЕНИЕ ТАБЛИЦ

## **Заполнение таблиц данными**

На рисунках 3-9 представлены значения, помещенные в таблицы при заполнении.

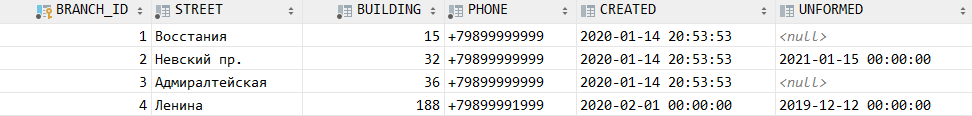


Рисунок 3. Заполнение таблицы «BRANCH»

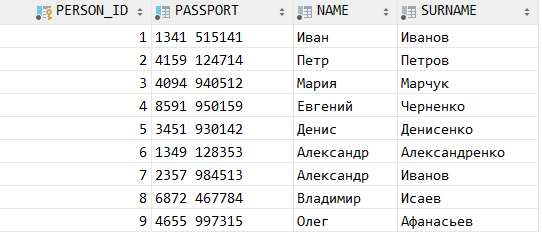


Рисунок 4. Заполнение таблицы «PERSON»

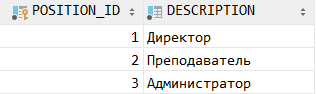


Рисунок 5. Заполнение таблицы «POSITION»

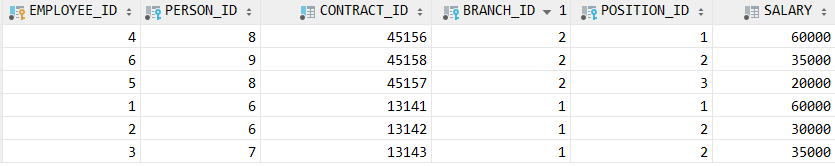


Рисунок 6. Заполнение таблицы «EMPLOYEE»



Рисунок 7. Заполнение таблицы «COURSE»

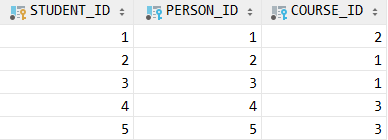


Рисунок 8. Заполнение таблицы «STUDENT»

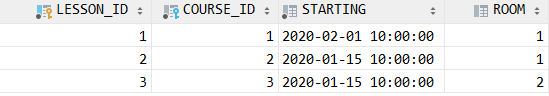


Рисунок 9. Заполнение таблицы «LESSON»

## 5 ВЫПОЛНЕНИЕ SQL – ЗАПРОСОВ

Основное средство для получения информации из базы данных — это запросы. Запрос – это средство извлечения данных из таблиц и предоставления их пользователю.

1. Список филиалов с указанием их расположения и директоров.

SELECT

BRANCH\_ID, BRANCH.STREET, BRANCH.BUILDING, PERSON.NAME, PERSON.SURNAME, POSITION.DESCRIPTION

FROM

BRANCH

INNER JOIN EMPLOYEE USING (BRANCH\_ID)

INNER JOIN POSITION USING (POSITION\_ID)

INNER JOIN PERSON USING (PERSON\_ID)

WHERE

POSITION.DESCRIPTION = 'Директор';

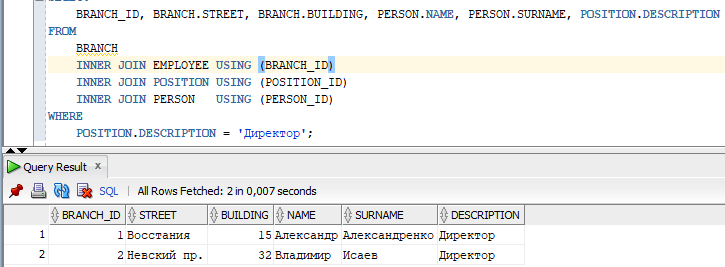


Рисунок 10. Результат запроса 1.

1. Список сотрудников указанного филиала с указанием должностей его сотрудников и их зарплат.

SELECT

EMPLOYEE.BRANCH\_ID, PERSON.NAME, PERSON.SURNAME, POSITION.DESCRIPTION, EMPLOYEE.SALARY

FROM

EMPLOYEE

INNER JOIN PERSON USING (PERSON\_ID)

INNER JOIN POSITION USING (POSITION\_ID)

WHERE

BRANCH\_ID = 1;

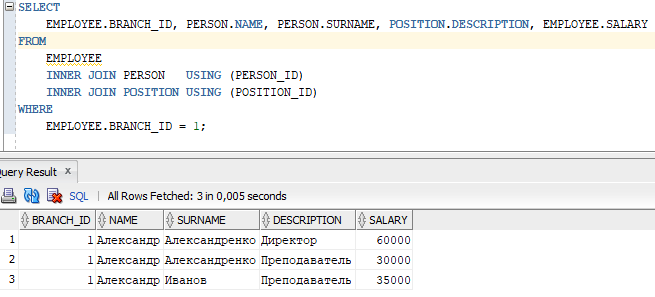


Рисунок 11. Результат запроса 2.

1. Список удаленных и вновь образованных филиалов за указанный период времени.

SELECT

\*

FROM

BRANCH

WHERE

CREATED > TO\_DATE('01.01.2010','dd.mm.yyyy')

AND UNFORMED < TO\_DATE('01.01.2030','dd.mm.yyyy')

AND CREATED > UNFORMED;

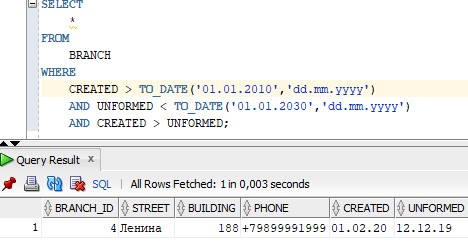


Рисунок 12. Результат запроса 3.

1. Список филиалов, фонд заработной платы которых превышает средний фонд зарплаты по всем филиалам.

SELECT

BRANCH\_ID, SUM(EMPLOYEE.SALARY)

FROM

EMPLOYEE

GROUP BY

BRANCH\_ID

HAVING

SUM(EMPLOYEE.SALARY) > (

SELECT

AVG(SUM(EMPLOYEE.SALARY))

FROM

EMPLOYEE

GROUP BY BRANCH\_ID);

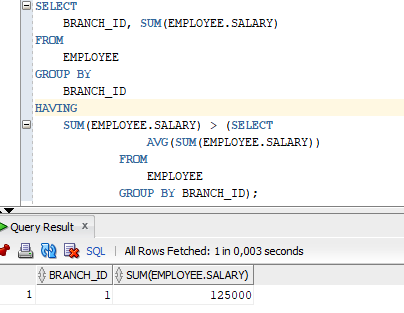


Рисунок 13. Результат запроса 4.

1. Список сотрудников, которые работают в разных филиалах.

SELECT

PERSON\_ID, PERSON.NAME, PERSON.SURNAME

FROM

PERSON

INNER JOIN EMPLOYEE USING (PERSON\_ID)

GROUP BY

PERSON\_ID, PERSON.NAME, PERSON.SURNAME

HAVING

COUNT(EMPLOYEE.POSITION\_ID) > 1;

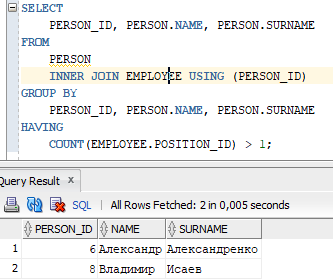


Рисунок 14. Результат запроса 5.

## ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы была разработана информационная система школы иностранных языков. Была проанализирована предметная область, создана логическая и физическая модель базы данных. Было выполнено прямое проектирование в программной среде Oracle SQL Developer. Таблицы были заполнены данными. Были построены пять SQL запросов, соответствующих заданию.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Карпова, Т.С. Базы данных: модели, разработка, реализация / Т.С. Карпова – СПб.: Питер, 2001 – 304с.
2. Онлайн документация базы данных Oracle [Интернет-ресурс] URL: https://docs.oracle.com/en/database/ (дата обращения: 02.01.2020)
3. Липанова, И.А. Управление данными: конспект лекций / И.А. Липанова – СПб.: СПБГУТ, 2016 – 45с.
4. Маклаков, С.В. Создание информационных систем с AllFusion Modeling Suite – М.: ДИАЛОГ-МИФИ, 2003 – 432с.
5. Мишра С., Бьюли А. Секреты Oracle SQL – пер. с англ. – СПб.: Символ-Плюс, 2003 – 386с.

## ПРИЛОЖЕНИЕ А

**CREATE TABLE BRANCH(**

**BRANCH\_ID NUMBER GENERATED AS IDENTITY PRIMARY KEY,**

**STREET VARCHAR(30) NOT NULL,**

**BUILDING NUMBER NOT NULL,**

**PHONE VARCHAR(20) NOT NULL,**

**CREATED DATE DEFAULT CURRENT\_DATE,**

**UNFORMED DATE**

**);**

**CREATE TABLE PERSON(**

**PERSON\_ID NUMBER GENERATED AS IDENTITY PRIMARY KEY,**

**PASSPORT VARCHAR(12) NOT NULL UNIQUE,**

**NAME VARCHAR(30) NOT NULL,**

**SURNAME VARCHAR(30) NOT NULL**

**);**

**CREATE TABLE POSITION(**

**POSITION\_ID NUMBER GENERATED AS IDENTITY PRIMARY KEY,**

**DESCRIPTION VARCHAR(30) NOT NULL UNIQUE**

**);**

**CREATE TABLE EMPLOYEE(**

**EMPLOYEE\_ID NUMBER GENERATED AS IDENTITY PRIMARY KEY,**

**PERSON\_ID NUMBER REFERENCES PERSON(PERSON\_ID) NOT NULL,**

**CONTRACT\_ID NUMBER NOT NULL UNIQUE,**

**POSITION\_ID NUMBER REFERENCES POSITION(POSITION\_ID) NOT NULL,**

**SALARY NUMBER NOT NULL**

**);**

**CREATE TABLE COURSE(**

**COURSE\_ID NUMBER GENERATED AS IDENTITY PRIMARY KEY,**

**LANGUAGE VARCHAR(20) NOT NULL,**

**BRANCH\_ID NUMBER REFERENCES BRANCH(BRANCH\_ID) NOT NULL,**

**YEAR\_SINCE NUMBER NOT NULL**

**);**

**CREATE TABLE STUDENT(**

**STUDENT\_ID NUMBER GENERATED AS IDENTITY PRIMARY KEY,**

**PERSON\_ID NUMBER REFERENCES PERSON(PERSON\_ID) NOT NULL UNIQUE,**

**COURSE\_ID NUMBER REFERENCES COURSE(COURSE\_ID) NOT NULL**

**);**

**CREATE TABLE LESSON(**

**LESSON\_ID NUMBER GENERATED AS IDENTITY PRIMARY KEY,**

**COURSE\_ID NUMBER REFERENCES COURSE(COURSE\_ID) NOT NULL,**

**EMPLOYEE\_ID NUMBER REFERENCES EMPLOYEE(EMPLOYEE\_ID) NOT NULL,**

**STARTING DATE NOT NULL,**

**ROOM NUMBER**

**)**