**ПРОФЕСИОНАЛНА ГИМНАЗИЯ ПО МЕХАНОТЕХНИКА, ЕЛЕКТРОТЕХНИКА, ТЕЛЕКОМУНИКАЦИИ** 

**И ТРАНСПОРТ – ШУМЕН**

**ДИПЛОМНА РАБОТА**

**НА**

**Селим Сеферов 12a КЛАС**

**СПЕЦИАЛНОСТ : ПРИЛОЖЕН ПРОГРАМИСТ**

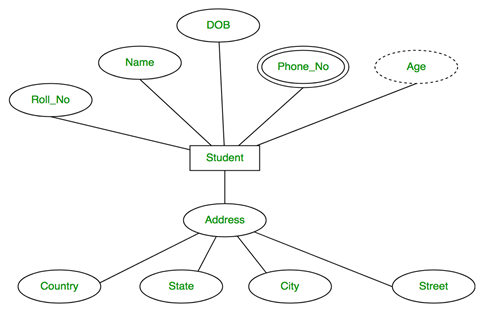
**ТЕМА:**

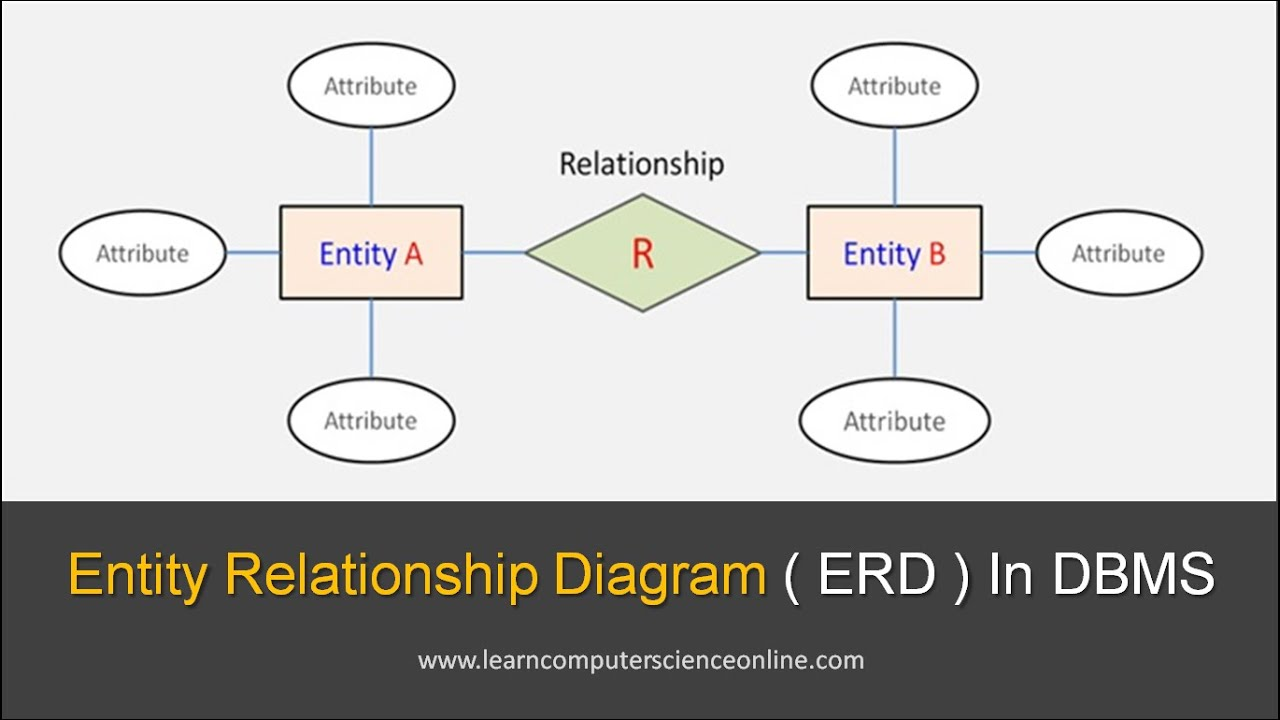
**Разработка на информационна система „Домоуправител“**

Изговил: ................... Ръководител: ...................

/Селим Сеферов 12А/ /инж. Васил Бояджиев/

1. Етажна собственост
2. Избор на тип база за изграждане на информационна система „Домоуправител“
3. Запознаване с моделите данни в базите данни

* За да бъдат удобни за обработване, данните в компютърната памет трябва да бъдат добре структурирани.
* През 1959 г. се формира международна работна група CODASYL (COmmittee on DAta SYstems Languages, Комитет по езиците за работа с данни). Целта на групата е, да работи върху проблемите на по-широкото използване на компютрите в различните сфери на човешката дейност. Първият резултат от работата на групата е специфицирането на езика COBOL (COmercial Business-Oriented Language, Език за обработка на бизнес данни).
* Възможностите на COBOL за структуриране на данни не могат да задоволят нуждите на изграждащите се ИС и затова специалистите продължават да работят за формулиране на универсални теоретични модели за структуриране на данните.
* Релационен (Табличен) модел
* Основа на релационния модел е двумерният масив – данните са организирани в редове и стълбове. Всички елементи са от един и същи тип, което ограничава възможностите за представяне на данните. Данните в една ИС могат да са от различни типове – текстови, числови, дати, изображения и др. В езиците за ООП този недостатък се преодолява до някаква степен с използването на класове от обекти.
* През 1970 г. Едгар Франк Код публикува своят модел, наречен релационен (табличен), според който данните се групират според типа им и за всяка група еднотипни данни се създава таблица. Така една ИС се състои от много на брой таблици, всяка от които е с еднотипни данни.
* Йерархичен модел
* Йерархии се срещат навсякъде в живота – в организацията на файловете в ОС, в холдингите, в университетите и т.н.
* През 1960 г. компанията IBM разработва йерархичния модел на данните. На негова основа компанията имплементира и разпространява чрез произвежданите от нея компютри IBM360, софтуер IMS (Information Management System) за разработване на ИС с йерархична организация на данните.
* С появата на модела на Код се вижда, че йерархичните структури сравнително просто се симулират в релационния модел. Затова йерархичния модел не получава широко разпространение и такива БД са рядкост.
* Мрежов (Графов) модел
* Графовите структури са най-сложната възможност за организация на данните. Имплементацията е още по-трудна, защото всеки обект от графа може да е свързан с всеки от останалите.
* С все по-широкото и разностранно използване на компютрите се оказва, че има области за които структурирането на данните е в графов модел.
* В днешно време се разработват системи за управление на графови бази от данни с най-различни приложения.
* ER (Entity-Relationship) модел
* Съвременен модел за работа с бази данни е ER-модела (Entity-Relationship model, отношения между обекти). Моделът е създаден от Питър Чен през 1976г. и е разработен за да се избегнат някои недостатъци на класическите модели.
* Спецификациите в този модел се представят в графичен вид с ER-диаграми.
* Данните са моделирани в таблици, наречени ядро на базата;
* Таблиците се състоят от редове/кортежи и колони/атрибути;
* Между таблиците има връзки;



1. Видове бази данни

* Релационни
* Релационните бази данни са един от най-разпространените видове бази данни и се използват от десетилетия. Те използват таблична структура с редове и колони за съхраняване и управление на данни. Данните са организирани в таблици, където всяка таблица представлява тип обект или обект, а редовете представляват отделни екземпляри на този обект. Релационните бази данни използват SQL (Structured Query Language) за заявки и манипулиране на данни.
* Нерелационни (NoSQL)
* NoSQL базите данни не използват традиционна таблична структура и могат да обработват неструктурирани или полуструктурирани данни. Те са проектирани да обработват големи обеми данни и могат да бъдат по-гъвкави и мащабируеми от релационните бази данни. NoSQL базите данни използват различни модели на данни, като базирани на документи, ключ-стойност, семейство колони и базирани на графики.
* Обектно-релационен
  + Обектно-релационен бази данни съхраняват данни като обекти, които могат да съдържат както данни, така и методи. Те са проектирани да работят с обектно-ориентирани езици за програмиране, като Java или C++. Обектно-ориентираните бази данни позволяват сложни връзки между данните и могат да бъдат по-ефективни за определени типове приложения.
* Йерархични
  + Йерархичните бази данни организират данните в дървовидна структура, като всеки запис има един родителски запис и един или повече дъщерни записи. Те често се използват в наследени системи и не са толкова гъвкави, колкото други видове бази данни.
* Мрежови
  + Мрежовите бази данни позволяват по-сложни взаимоотношения между данните чрез използване на подобна на графика структура с възли и ръбове. Те са подобни на йерархичните бази данни, но позволяват множество родителски и дъщерни записи. Мрежовите бази данни бяха популярни през 60-те и 70-те години на миналия век, но до голяма степен бяха заменени от други видове бази данни.
* Графични
  + Графичните бази данни използват графови структури за представяне и съхраняване на данни, като възлите и ръбовете представляват обекти и връзки между тях. Те са особено полезни за данни, които имат много сложни връзки, като социални мрежи или двигатели за препоръки. Графичните бази данни могат да бъдат по-ефективни от други типове бази данни за определени типове заявки.
* Документно ориентирани
  + Базите данни, ориентирани към документи, съхраняват и извличат данни като документи, обикновено под формата на JSON или XML файлове. Те често се използват за уеб приложения, системи за управление на съдържание и мобилни приложения. Базите данни, ориентирани към документи, могат да бъдат по-гъвкави от релационните бази данни и могат да обработват полуструктурирани данни.
* Time-series бази данни
  + Time-series са бази данни, които се променят с времето, като например данни от сензори или данни от финансовите пазари. Те са оптимизирани за съхранение и извличане на големи обеми данни в хронологичен ред. Базите данни с времеви редове стават все по-важни, тъй като Интернет на нещата (IoT) генерира все повече и повече данни с времеви редове.
* Пространствени
  + Пространствените бази данни съхраняват и управляват пространствени данни, като карти или геопространствени данни. Те се използват в различни приложения, като ГИС (географски информационни системи), базирани на местоположение услуги и мониторинг на околната среда. Пространствените бази данни използват специализирани алгоритми и структури от данни за ефективно съхраняване и заявки за пространствени данни.
* In-memory бази данни
  + Базите данни в паметта съхраняват данни в RAM вместо на твърд диск, което позволява по-бързо време за достъп. Те често се използват в приложения, които изискват бързи скорости на четене и запис, като анализи в реално време или системи с голям обем транзакции. Базите данни в паметта могат да бъдат по-скъпи от други видове бази данни поради цената на RAM.

1. Нива на абстракция
   * Абстрахиране – подход за проектиране, реализиране и управление на сложни системи.
   * Специалистите, участващи в процеса се разделят на категории, според профила си. Специалистите от всяка категория определят как ще изглежда системата от тяхна гледна точка и според техните компетенции. Останалите елементи се игнорират, т.е. специалистите се абстрахират от тях.
   * Сложният проект се разбива на нива на абстракция, с всяко от които да се заеме екип от специалисти, чиято компетентност съответства на нивото на абстракция

* Ниво на крайния потребител
  + Най – високото ниво на абстракция.
  + Възложителят се нуждае от софтуер, който да задоволи неговите нужди от информация и да подпомага работата му.
  + Възложителят не познава инструментариума, с който да се разработи ИС, не познава моделите на структуриране на данни, не познава и СУБД, с която да се реализира системата. Той описва нуждите си от информация на естествен език.
  + Възможно е обаче описанието на естествен език да не е достатъчно, за да стартира проектирането на системата.
* Ниво на бизнес анализатора
  + Специалистите по бизнес анализ:
  + Имат способността да вникнат в същността на работата на крайния потребител, да анализират процесите и да определят по-точно нуждите от информация.
  + Владеят инструменти за формализирано описание на информационните процеси и могат да преведат описанието от естествен език в език на инструмента за формализирано описание на ИС.
  + ЕR – моделирането е един такъв инструмент за формализирано представяне на информационните процеси и нужди на крайния потребител.
* Ниво на разработчика
  + Специалисти, които познават инструментите за реализация на базата данни и интегрирането й в ИС. Те превеждат ER – модела на езика на СУРБД, като съпоставят на всеки обект от ER – модела таблица с атрибутите му.
  + Планират и таблиците за осъществяване на връзки между обекти.
  + Добавят атрибути, които бизнес анализатора не е предвидил, защото се е абстрахирал от тях.
  + Изготвят Инструкция за крайния потребител, в която посочват как той да въвежда данни и как да постъпи за получаване на нужните справки.
  + Разработчиците също се абстрахират от някои елементи на системата, които са задължение на системните администратори.
* Аномалии
  + Ако една БД не е нормализирана възникват аномалии, които се проявяват като осиротели записи, сочещи към несъществуващи данни или като невъзможност за добавяне на запис.
  + АНОМАЛИЯ НА ВМЪКВАНЕТО;
  + АНОМАЛИЯ НА ОБНОВЯВАНЕТО;
  + АНОМАЛИЯ НА ИЗТРИВАНЕТО.

1. Избираме релационни бази данни (SQL)
   * Базите от данни се управляват от специален софтуер, който ще наричаме  
     СУБД (система за управление на бази от данни). СУБД представлява мощен  
     инструментариум за създаване и ефективно управление на големи обеми от  
     данни. Данните трябва да се поддържат и съхраняват за толкова време за колкото  
     е необходимо. Освен това, те трябва да се предпазват от неправилен достъп, който  
     може да наруши целостта им (integrity), както и от неправомерен достъп  
     (security).  
     Примери:  
     MS SQL Server, Oracle Database, IBM DB2, MySQL, PostgreSQL
   * Ще въведем и още две нови понятия, свързани със структурата на базата от  
     данни:  
     1. Data Definition Language (DDL) се използва за създаване на схема  
     (описание) на данните. Чрез него абстрактно се дефинира структурата на  
     информацията в базата от данни и се дефинират релации между различните  
     компоненти. Основна грижа на DDL също е да дефинира ограниченията за  
     цялостност на информацията (с цел постигане на интегритет) и да се проектира  
     сигурността на базата от данни.  
     2. Data Manipulation Language (DML) са езиковите средства за обработка на  
     данните. Най-често те се разделят на вътрешни (езика използван от сървъра за  
     базата от данни, например SQL) и външни (езиците за програмиране, например  
     C/C++, Java, и т.н.). Както подсказва самото име, чрез тези езици се подават  
     команди, чрез които сървъра обработва данните.
   * Езикът SQL най-общо казано се използва са създаване на бази от данни,  
     създаване на таблици и връзките между тях, вмъкване на информация, извличане  
     на информация, промяна на информация и изтриване на информация от таблици  
     в база от данни. Въпреки съществуването на официален стандарт на езика, почти  
     всяка система за управление на бази от данни има своя собствена имплементация.  
     Почти винаги се спазват основните правила в езика, но често се добавя  
     допълнителна функционалност, която е специфична за използваната СУБД.  
     SQL се базира на последователност от заявки (команди). Когато част от тези  
     заявки са зависими една от друга говорим за понятието ”транзакция”. Вече  
     споменахме за DDL и DML. Ето как най-общо са въведени тези понятия чрез езика  
     SQL:
   * 1. Data Definition Language:  
     - CREATE DATABASE – създава база от данни  
     - ALTER DATABASE – променя дефиницията на база от данни  
     - DROP DATABASE – изтрива база от данни  
     - CREATE TABLE – създава таблица в база от данни  
     - ALTER TABLE – променя дефиницията на таблица в база от данни  
     - DROP TABLE – изтрива таблица от база от данни  
     - CREATE INDEX – създава индекс  
     - DROP INDEX – изтрива индекс  
     2. Data Manipulation Language:  
     - INSERT – вмъква информация в база от данни  
     - SELECT – извлича информация от база от данни  
     - UPDATE – променя информация в база от данни  
     - DELETE – изтрива информация от база от данни
   * Въпреки, че SQL би трябвало да е унифициран език, всяка система за  
     управление на бази от данни се различава от другите. Много рядко се спазва  
     истинския ANSI стандарт на SQL и често има съществени различия.  
     Както споменахме по-горе, релационните бази от данни съхраняват  
     информацията в таблици. Таблиците се състоят от редове, като всеки ред е един  
     запис в базата, а клетките от таблицата са отделните полета в него. Таблиците  
     имат по една или повече колони, като всяка колона има име и тип. Колоните  
     определят съответните полета в записите от базата. Всяка колона определя типа на  
     поле от запис или типа данни, с които можем да запълним конкретните полета от тази  
     колона.
2. Реализация на проекта