Изпитна тема № 9: Бази данни

Заявки за извличане и промяна на данни. Вложени заявки. Обединение на заявки. Съединения на заявки за извличане на данни от бази данни. Извличане на части от таблица/и и задаване на псевдоними.

Таблиците в базите данни могат да бъдат оприличени на обекти, които съхраняват данни. В таблицата данните са организирани във формата на колони и редици. Редиците представляват информация за уникален запис (record), а колоната съответства на едно поле от редицата.

За извличане на данни от таблицата се използва запазената дума **SELECT.** След това се изреждат със запетая имената на колоните, от които искаме данни. След имената се поставя запазената дума **FROM** и съответно името на таблицата, от която ще бъдат извлечени данните. Ако има нужда от филтрация на данните се използва клаузата **WHERE**.

**SELECT** first\_name,

last\_name

**FROM** Persons;

За промяна на данни в дадена таблица се използва запазената дума **UPDATE.** Първо се посочва таблицата, чиито данни ще бъдат променени. След това следва **SET** клаузата, в която се изреждат колоните, които подлежат на промяна и съответно новите им стойности. При нужда от условие, кои колони да бъдат променяни се използва **WHERE** клаузата. Така ще се променят данните в колоните, само ако стойностите им отговарят на зададеното условие.

**UPDATE** Persons,

**SET** first\_name = ‘Azis’

**WHERE** first\_name = ‘Vasil’;

**FROM** Persons;

Вложените заявки се използват за прецизно филтриране на данните или като източник на други данни за други заявки. Най-често вложените заявки се намират в **WHERE** клаузата като в условието се използва стойността, която са върнали. Например, ако имаме таблица с две колони име на ученик (name) и резултат (result), с вложени заявки можем да получим имената на всички ученици, които имат резултат, по-висок от средния за цялата таблица.

**SELECT** name

**FROM** students

**WHERE** result > (

**SELECT AVG**(result)

**FROM** students);

**FROM** Persons;

Обединените заявки представляват комбинация от резултатите от 2 заявки за извличане на информация. За такива заявки се използват операторите **UNION** и **UNION ALL.** Операторът **UNION** комбинира двата резултата и премахва дубликатите, като по този начин гарантира, че всеки ред ще е уникален. **UNION ALL** също комбинира резултатите, но не премахва повтарящите се елементи.

Операторът **SELECT** е ограничен да връща данни само от една таблица. Често се налага информацията, която трябва да бъде върната да се намира в повече от една таблица. Процеса по връщане на информация от 2 или повече таблици се нарича свързване на таблици или съединение (join). За свързване на таблици се използват клаузите **INNER JOIN**, **LEFT JOIN**, **RIGHT JOIN** и **FULL OUTER JOIN**.

Най-често използваната клауза е **INNER JOIN**. Тя слива таблиците по някаква обща колона и връща записите, които имат съвпадащи стойности и в двете таблици

A picture containing circle, diagram, sketch, design

Description automatically generated

В следния пример имаме две таблици: хора (persons) и адреси (addresses). В таблицата persons има колони: id (първичен ключ), name и address\_id (външен ключ). В таблицата addresses има колони: id (първичен ключ), town\_name и street\_name. За да извлечем имената на всички хора и съответно града, в който живеят трябва да използваме **INNNER JOIN**. Клаузата се поставя след запазената дума **FROM**, като в нея се описва с коя ще бъде съединението и условието, според което ще се съединят.

**SELECT** name,

a.town\_name

**FROM** persons AS p

**INNER JOIN** addresses a ON a.id = p.id;

Извличането на части таблица се използва, за да извлечем само данните, които са ни необходими във удобен формат. За да задаваме имена на колоните, които са обединение на една или повече колони се използват така наречените псевдоними (alias). Те се дефинират, чрез запазената дума **AS**. По този начин се задава име на колоната или на израза, докато заявката се изпълнява. Използва се веднага след оригиналното име на колоната или израза. Ако псевдонима съдържа празни места, той трябва да бъде ограден в единични кавички. Например имаме таблицата persons, която има 3 колони – id(първичен ключ), first\_name и last\_name. Ако искаме да получим пълното име на даден човек, трябва да конкатенираме двете колони да дефинираме псевдоним.

**SELECT** first\_name + ‘ ’ + last\_name AS ‘Full name’

**FROM** persons;

Обобщен план (Опорни думи)

1. Заявки за извличане и промяна на данни
   * SELECT – Имена на колони. FROM. WHERE.
   * UPDATE- Име на таблица. SET име на колона и нова стойност. WHERE.
2. Вложени заявки – филтриране на данни, източник на данни, най-често в WHERE клаузата. Пример – Всички хора с резултат над средния.
3. Обединение на заявки
   * UNION – обединява две заявки, премахва дубликати.
   * UNION ALL – комбинира две заявки, не премахва дубликати.
4. Съединения на заявки за извличане на данни от бази данни – SELECT е ограничен до 1 таблица, процеса по извличане на данни от 2 или повече – свързване на таблици
   * INNER JOIN – слива по обща колона, връща съвпадащи стойности. Диаграми кръгчета – в средата оцветено. Пример – хора, адреси, извличане на име и адрес на всеки човек.
   * LEFT JOIN
   * RIGHT JOIN
   * FULL OUTER JOIN
5. Извличане на части от таблица/и и задаване на псевдоними.
   * Извличане на необходимите данни в удобен формат
   * Псевдоними (alias) – задаваме име на колона/израз по време на изпълнение на заявката. AS. Използва се веднага след оригиналното име. При празно място в името се огражда с ‘’. Пример first\_name + last\_name.

Казус

Напишете заявка, която показва всички служители подредени по азбучен ред.

**SELECT** FirstName,

MiddleName,

LastName

**FROM** Employees

**ORDER BY** FirstName **ASC**;

Намерете първите 5 служителя(First Name, Hire date), назначени след „06-02-2000“. Подредете ги по първо име в азбучен ред.

**SELECT** **TOP** 5 FirstName,

HireDate

**FROM** Employees

**WHERE** HireDate > ’06-02-2000’

**ORDER BY** FirstName **ASC**;

Напишете заявка, която селектира и създава електронни пощи на потребители, съдържащи техните първи имена, фамилия и се добави домейна ‘bulgaria.bg‘. Преименувайте колоната на ‘Emails’ и ги подредете по първо име в азбучен ред. Използвайте функция.

**SELECT** **CONCAT** (FirstName,

‘.’,

LastName,

‘@bulgaria.bg ’) AS Emails

**FROM** Employees

**ORDER BY** FirstName **ASC**;

Напишете SQL Query, което намира всички First Name, Middle Name, Last Name, на които First Name започват с буквата “A” и на всички, които Middle Name им е “F”. Съединете заявките и ги подредете по Middle Name в азбучен ред.

**SELECT** **TOP** e.FirstName,

e.MiddleName,

e.LastName

**FROM** Employees **AS** e

**WHERE** e.FirstName **LIKE** ‘A%’

**UNION**(

**SELECT** emp.FirstName,

emp.MiddleName,

emp.LastName

**FROM** Employees **AS** emp

**WHERE** emp.MiddleName **LIKE** ‘F’)

**ORDER BY** e.MiddleName **ASC**’;

Напишете заявка, която извлича име(FirstName), фамилия(LastName), град(Town) и адрес(Address). Сортирайте ги по име(FirstName) в азбучен ред и покажете първите 10 служителя. Преименувайте колоните.

**SELECT** **TOP** 10 e.[FirstName],

e.[LastName],

t.[Name] AS [Town],

a.[AddressText] AS [Address]

**FROM** Employees **AS** e

**INNER JOIN** Addresses **AS** a

**ON** e.AddressID = a.AddressID

**INNER JOIN** Towns **AS** t

**ON** a.TownID = t.TownID

**ORDER BY** e.FirstName **ASC**;