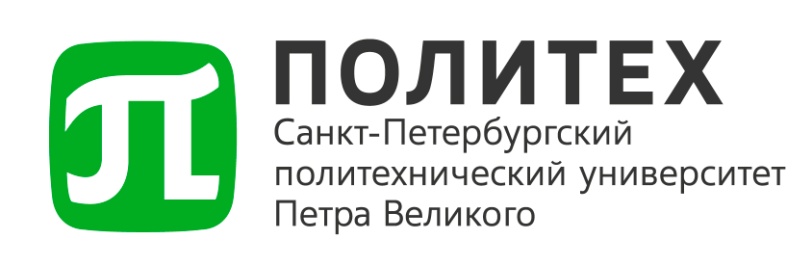
ФЕД­­ЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**



**Курсовая работа**

**«База данных печатных изданий**»

**по дисциплине «Управление данными в приборостроении"**

Выполнил

студент гр.23535/2 Летюшев А. И.

Преподаватель Малыхина Г. Ф.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2018 г.

Санкт-Петербург

2018

Оглавление

Введение 1

1. Моделирование данных 2

1.1 E/R-модель данных3

1.2 ODL-модель 4

2. База данных Oracle 7

2.1 Создание базы данных печатных изданий 7

2.2 Запросы к базе данных печатных изданий 8

3. Триггеры базы данных 18

3.1 Триггер уровня строки, с псевдозаписями 18

3.2 Триггер с предикатами if-then20

3.3 Триггер с условием when 22

4. Программирование клиента базы данных23

4.1 Первая транзакция23

4.2 Вторая транзакция25

Заключение 26

Список источников 27

Введение

В данной работе будут рассмотрены моделей баз данных, реляционные модели и реляционная алгебра запросов, представлены основные методы и операторы языка запросов SQL и PL\SQL, элементы управления баз данных с помощью средств JDBC.

В ходе работы будут изучены системные аспекты языка структурированных запросов SQLи среда программирования, интерфейс SQL и главного языка программирования, применение триггеров и управление транзакциями.

Основной задачей ставится освоение методов поэтапной разработки базы данных и управление данными с помощью функций языка SQL и PL\SQL.

Актуальность проделанной работы состоит в изучении современных средств программирования в СУБД Oracle и ознакомлении с основными концепциями языка запросов SQL.

1. Моделирование данных

Проектирование базы данных начинается с построения моделей данных, разработки их организации и представления. Например, необходимо выполнить описание хранимой информации в виде объектов, содержащих определенный кортежи данных и имеющих уникальные ключевые атрибуты, а также связи с другими объектами. Традиционно выполняют построение E/R-модели или объектно-ориентированной ODL модели.

Схема E/R-модели содержит сущности, атрибуты и связи. E/R-модель позволяет представить 3 типа связей сущностей: связи один к одному, связь одного объекта со многими и множественные связи между объектами сущностей.

1.1 E/R-модель данных

Схема E/R-модели содержит сущности, атрибуты и связи. E/R-модель позволяет представить 3 типа связей сущностей: связи один к одному, связь одного объекта со многими и множественные связи между объектами сущностей.

Journal

Author

Printing house

Publishing house

Client

Write

Order

Publish

Print

1.2 ODL-модель

Объектно-ориентированная модель данных, для описания которой используется ODL, позволяет выполнить прямой переход к выражениям ООСУБД. Что делает этот моделирование данных на этом языке важным этапом в разработке базы данных

Interface Cloud (key name)

{

Attribute string name;

Attribute string tech;

Attribute string type;

Attribute string company;

}

relationship resource center use;

inverse resource center::obey;

Interface Resource center (key city)

{

Attribute string director;

Attribute string city;

Attribute int numb nodes;

}

relationship cloud obey;

inverse cloud::use;

relationship set <cluster> rule;

inverse cluster::conform;

relationship set <data storage> save;

inverse data storage::control;

relationship set<calculation> regulate;

inverse calculation::perform;

Interface Cluster (key ID, address)

{ Attribute int ID;

Attribute struct add {

string street;

int house;

}address;

Attribute int life time;

Attribute int numb servers;

}

relationship resource center conform;

inverse resource center::rule;

relationship set <server> work;

inverse server::keep;

Interface Server (key serial\_numb)

{

Attribute int serial\_numb;

Attribute float mem\_size;

Attribute float power;

}

relationship cluster keep;

inverse cluster::work;

Interface Data storage (key cost, space)

{

Attribute float cost;

Attribute float space;

Attribute float available;

}

relationship resource center control;

inverse resource center::save;

relationship account belong;

inverse account::make;

Interface Сalculation (key price)

{

Attribute float price;

Attribute float speed;

Attribute float consumption;

Attribute string client;

}

relationship resource center perform;

inverse resource center::regulate;

relationship set<account> practice;

inverse account::own;

Interface Account (key login, password)

{

Attribute string login;

Attribute string password;

Attribute int subs;

Attribute struct data {

int year; month, day;

} d.o.b.;

}

relationship data storage make;

inverse data storage belong;

relationship set <calculation> own;

inverse calculation::practice;

База данных находится в нормальной форме Бойса-Кодда (НФБК), так как все атрибуты являются простыми, все не ключевые атрибуты отношений зависят от ключевых, нет транзитивных зависимостей и ключи некоторых отношений являются составными.

2. База данных Oracle

В настоящее время наиболее широко распространенные реляционные базы данных используют язык запросов SQL. Этот язык строится на основе правил реляционной алгебры, но некоторые важные свойства выходят далеко за его пределы, поэтому далеко не все запросы и функции этого языка представимы с помощью операций теории множеств, дискретной математики и реляционных моделей.

2.1 Создание базы данных облачных хранилищ и инициализация таблиц методом insert

Оператор CREATE TABLE позволяет создавать новые таблицы данных, которые могут быть заполнены данными с помощью оператора INSERT или удалены с помощью DROP. SQL-запросы основаны на предложении SELECT-FROM-WHERE, применяемое к одному или объединению отношений

Drop table Printing\_house;

CREATE TABLE Printing\_house (name\_print varchar2(100) primary key, address varchar2(100), index1 int, phone\_number varchar2(100), typografy\_number int);

Drop table Journal;

CREATE TABLE Journal (title varchar2(100) primary key, year\_of\_creation int, number\_of\_issues int, edition int, number\_of\_pages int, production\_number int, name\_print varchar2(100) REFERENCES Printing\_house(name\_print));

Drop table Publishing\_house;

CREATE TABLE Publishing\_house (name\_publ varchar2(100), address varchar2(100), index2 int, phone\_number varchar2(100), publish\_number int,

Drop table Authors;

CREATE TABLE Authors (name varchar2(100) primary key, place\_of\_work varchar2(100), specialization varchar2(100), qualification int, author\_number int);

Drop table Clients;

CREATE TABLE Clients (client varchar2(100) primary key, address varchar2(100), phone\_number varchar2(100), clients\_number int);

drop table WriteJornal;

create table WriteJornal (name varchar2(100) REFERENCES Authors(name), title varchar2(100) REFERENCES Journal(title));

drop table OrderClient;

create table OrderClient (client varchar2(100) REFERENCES Clients(client), title varchar2(100) REFERENCES Journal(title));

2.2 Запросы к базе данных облачных хранилищ

1.Простой запрос SELECT-FROM-WHERE к одной таблице – 1 запрос.

Вывести название журнала, у которого тираж составляет 90000 экземпляров.

SELECT title

FROM Journal

WHERE edition = 90000;

2. Простой запрос SELECT-FROM-WHERE к двум таблицам – 1 запрос.

Вывести названия имена авторов и названия журналов, основанных ранее 1980 года.

SELECT Journal.title, name FROM Journal, WriteJornal WHERE year\_of\_creation> 1980 and Journal.title = WriteJornal.title;

|  |  |
| --- | --- |
| **TITLE** | **NAME** |
| Functional Ecology | Charles Gibson |
| Teoreticheskaya i prikladnaya ekologiya | Newton Melissa |
| Functional Ecology | Gleb Sidorov |

3. Простой запрос SELECT-FROM-WHERE к трем таблицам – 1 запрос.

Вывести название журнала, а также адреса его печати и издательства.

SELECT Journal.title, Printing\_house.address Add\_Print, Publishing\_house.address Add\_Publ FROM Journal, Printing\_house, Publishing\_house WHERE Journal.name\_print=Printing\_house.name\_print and Publishing\_house.title = Journal.title;

|  |  |  |
| --- | --- | --- |
| **TITLE** | **ADD\_PRINT** | **ADD\_PUBL** |
| Journal of Ecology | 1867 Crestview Manor Indianapolis | 2203 Shobe Lane Vail |
| GEO | ul. Aviatorov, 69 | 0586 Crestview Manor Indianapolis |
| Nature | ul.Barminovskaya, 68 | 1703 Bond Street Newport |
| Ecology | ul.Barminovskaya, 68 | 5169 Junior Avenue Atlanta |
| Journal of Animal Ecology | 1867 Crestview Manor Indianapolis | 1111 Martha Street Yuma |
| Functional Ecology | ul.Barminovskaya, 68 | ul. Kompozitorov, 151 |
| Vokrug sveta | ul. Admirala Rudneva, 77 | pr.Engelsa , 12 |
| Yunyy naturalist | ul. Aviatorov, 69 | ul. Korablestroiteley, 25 |
| Teoreticheskaya i prikladnaya ekologiya | ul. Admirala Rudneva, 77 | ul. Nepokorennaya, 2 |
| Ecologiya cheloveka | ul. Aviatorov, 69 | ul. Ivana Fomina, 9 |

4. Запрос к одной таблице, умноженной на себя – 1 запрос.

Вывести названия журналов и соответствующих им клиентов.

select distinct A1.client, A2.title from OrderClient A1, OrderClient A2 where A1.title=A2.title and A1.client!=A2.client ORDER BY A2.title;

|  |  |
| --- | --- |
| **TITLE** | **CLIENT** |
| Journal of Ecology | The Hong Kong Polytechnic University |
| Journal of Ecology | Yale University |
| Teoreticheskaya i prikladnaya ekologiya | Mining University |
| Teoreticheskaya i prikladnaya ekologiya | Polytech University |
| Teoreticheskaya i prikladnaya ekologiya | Tor Books |

5. Запросы с подзапросами, использующие 2 таблицы (2 запроса) и 3 таблицы (2 запроса).

Вывести журналы, публикующиеся Дрофой.

Select Journal.title FROM Journal

WHERE Journal.title= (Select Publishing\_house.title

FROM Publishing\_house WHERE name\_publ = 'Drofa');

|  |
| --- |
| **TITLE** |
| Teoreticheskaya i prikladnaya ekologiya |

Вывести адрес всех печатей, работающих с журналами где номер продукции равен 23668.

Select address FROM Printing\_house

WHERE Printing\_house.name\_print= (Select Journal.name\_print

FROM Journal WHERE Journal.production\_number= 23668);

Вывести имена авторов, публикующихся Kniga.

SELECT WriteJornal.name From WriteJornal WHERE WriteJornal.title=

(Select Journal.title FROM Journal

WHERE Journal.title = (select Publishing\_house.title from Publishing\_house where name\_publ='Kniga'));

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **NAME** | | Newton Melissa |  |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |

Вывести клиентов, сотрудничающих с журналом Вокруг света и печатающихся по индексу 18565.

SELECT OrderClient.client From OrderClient WHERE OrderClient.title=

(Select Journal.title FROM Journal

WHERE Journal.name\_print= (select Printing\_house.name\_print from Printing\_house where index1=18565) and title='Vokrug sveta');

|  |
| --- |
| **CLIENT** |
| Future plc |

Каждый из запросов выполняется дважды используя слово из (EXISTS, IN) и/или из ( ALL, ANY). Все слова должны быть в запросах использованы – 4 запроса. Каждый запрос сделайте дважды, используя (EXISTS, IN), используя (ALL, ANY).

Найти номера телефонов клиентов, сотрудничающих с Журналом об экологии.

SELECT phone\_number from Clients where Clients.client IN(select OrderClient.client from OrderClient where OrderClient.title='Journal of Ecology');

Тоже самое, но через EXISTS.

SELECT phone\_number from Clients where EXISTS(select \* from OrderClient where OrderClient.title='Journal of Ecology' and Clients.client=OrderClient.client);

|  |
| --- |
| **PHONE\_NUMBER** |
| 10(20)365-02-88 |
| 53(60)145-20-99 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Найти печатные дома и их адреса, под ответственностью которых не лежат печать журналов тиражем менее 70000.

SELECT Printing\_house.name\_print, Printing\_house.address from Printing\_house where Printing\_house.name\_print <>ALL(select Journal.name\_print from Journal where edition <70000);

Тоже самое, но с использованием IN.

SELECT Printing\_house.name\_print, Printing\_house.address from Printing\_house where NOT Printing\_house.name\_print =ANY(select Journal.name\_print from Journal where edition <70000);

|  |  |
| --- | --- |
| **NAME\_PRINT** | **ADDRESS** |
| Skott | ul.Zhelyabova, 39 |
| Doubleday | 1255 Bond Street Newport |
| Conde Nast | 4192 Junior Avenue Atlanta |
| Tor Books | 3178 Leroy Lane Rapid City |
| Berlitz | 1648 Shobe Lane Vail |
| Khatber | ul. Shostakovicha, 39 |

6. Запросы теории множеств UNION, INTERSECT, EXCEPT(MINUS) – 3 запроса.

Вывести печатные дома, которые имеют заказы от журналов.

select Printing\_house.name\_print from Printing\_house INTERSECT select Journal.name\_print from Journal;

|  |
| --- |
| **NAME\_PRINT** |
| Drofa |
| Future plc |
| Nauka |
| Prosveshcheniye |

Вывести журналы, которые не имеют клиентов в издательстве.

Select Publishing\_house.title FROM Publishing\_house MINUS Select OrderClient.title FROM OrderClient;

|  |
| --- |
| **TITLE** |
| Ecologiya cheloveka |
| Ecology |
| Journal of Animal Ecology |
| Yunyy naturalist |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Вывести названия организаций связанные с журналом GEO.

Select Publishing\_house.name\_publ Firm FROM Publishing\_house where Publishing\_house.title='GEO' UNION Select Printing\_house.name\_print Firm FROM Printing\_house where Printing\_house.name\_print =(Select Journal.name\_print from Journal where title='GEO');

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |
| --- |
| **FIRM** |
| Drofa |
| Tokyopop |

7. Пользовательское представление из 3 таблиц. 1 запрос к представлению – 1 запрос.

Создать пользовательское представление, где связаны данные названия журнала, его тираж, названия его печатного дома и издательства.

CREATE VIEW Journ\_Publ\_Print AS (Select Journal.title, edition, Printing\_house.name\_print, Publishing\_house.name\_publ

FROM Journal, Printing\_house, Publishing\_house

WHERE Journal.title = Publishing\_house.title AND Printing\_house.name\_print = Journal.name\_print);

Вывести всю информацию о журналах, тираж которых больше 80000.

select \* from Journ\_Publ\_Print where edition>80000;

|  |  |  |  |
| --- | --- | --- | --- |
| **TITLE** | **EDITION** | **NAME\_PRINT** | **NAME\_PUBL** |
| Nature | 85000 | Prosveshcheniye | Simon & Schuster |
| Journal of Animal Ecology | 90000 | Future plc | Basic Books |
| Vokrug sveta | 100000 | Nauka | Impuls |

8. Внешние и внутренние соединения (JOIN ON, CROSS JOIN, NATURAL OUTER

JOIN,OUTER , NATURAL LEFT OUTER, NATURAL RIGHT OUTER ) – 6 запросов.

Вывести имена, и специальность авторов, работающих на журналы, клиентами которых является их рабочее место.

select name,specialization from Authors JOIN Clients ON place\_of\_work=client;

|  |  |
| --- | --- |
| **NAME** | **SPECIALIZATION** |
| Charles Gibson | Professor |
| Newton Melissa | Сandidate of Sciences |
| Oparin Maksim | Dean |

Вывести названия журналов и их авторов, которые печатаются Наукой.

SELECT Journal.title, name FROM Journal CROSS JOIN WriteJornal WHERE name\_print='Nauka' and Journal.title = WriteJornal.title;

Вывести названия журналов и связанные с ними имена авторов и клиентов.

SELECT title, name, client FROM WriteJornal NATURAL JOIN OrderClient where title='Journal of Ecology';

|  |  |  |
| --- | --- | --- |
| **TITLE** | **NAME** | **CLIENT** |
| Journal of Ecology | Charles Gibson | Yale University |
| Journal of Ecology | Charles Gibson | The Hong Kong Polytechnic University |
| Journal of Ecology | Allison Cook | Yale University |
| Journal of Ecology | Allison Cook | The Hong Kong Polytechnic University |
| Journal of Ecology | Petrova Maria | Yale University |
| Journal of Ecology | Petrova Maria | The Hong Kong Polytechnic University |
| Journal of Ecology | Oparin Maksim | Yale University |
| Journal of Ecology | Oparin Maksim | The Hong Kong Polytechnic University |

Вывести названия печатных домов, их номера и заказывающие у них журналы, номер которых начинается с 8.

SELECT Printing\_house.name\_print, phone\_number, Journal.title

FROM Printing\_house FULL JOIN Journal

ON Printing\_house.name\_print= Journal.name\_print where phone\_number LIKE '8%';

|  |  |  |
| --- | --- | --- |
| **NAME\_PRINT** | **PHONE\_NUMBER** | **TITLE** |
| Drofa | 8(916)015-95-46 | GEO |
| Prosveshcheniye | 8(812)633-22-80 | Nature |
| Prosveshcheniye | 8(812)633-22-80 | Ecology |
| Prosveshcheniye | 8(812)633-22-80 | Functional Ecology |
| Nauka | 8(622)528-39-77 | Vokrug sveta |
| Drofa | 8(916)015-95-46 | Yunyy naturalist |
| Nauka | 8(622)528-39-77 | Teoreticheskaya i prikladnaya ekologiya |
| Drofa | 8(916)015-95-46 | Ecologiya cheloveka |
| Khatber | 8(931)307-69-31 | - |
| Skott | 8(637)865-67-41 | - |
| Doubleday | 82(69)166-58-64 | - |
|  |  |  |

Вывести названия и номера организаций, одновременно являющихся и печатным домом и издательством.

Select distinct name\_publ Name, Publishing\_house.phone\_number

FROM Publishing\_house RIGHT JOIN Printing\_house ON name\_print=name\_publ;

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **NAME** | **PHONE\_NUMBER** | | - | - | | Drofa | 8(936)988-11-06 | |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Вывести имена авторов и их клиентов, где клиент является работодателем автора

Select distinct name, client

FROM Clients LEFT OUTER JOIN Authors ON place\_of\_work=client;

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |
| --- | --- |
| **NAME** | **CLIENT** |
| Charles Gibson | The Hong Kong Polytechnic University |
| Newton Melissa | Yale University |
| Oparin Maksim | Mining University |
| - | First page |
| - | State library |
| - | Tor Books |
| - | Doubleday |
| - | Berlitz library |
| - | Polytech University |
| - | Future plc |

9. Агрегация (SUM, COUNT, AVG, MIN, MAX) – 1 запрос.

Вывести названия всех фирм, скорость и стоимость вычислений, где скорость вычислений выше средней.

Select title, production\_number

FROM Journal

WHERE edition >ALL(Select AVG(edition) FROM Journal);

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |
| --- | --- |
| **TITLE** | **PRODUCTION\_NUMBER** |
| Journal of Ecology | 23668 |
| Nature | 53698 |
| Ecology | 78023 |
| Journal of Animal Ecology | 98201 |
| Vokrug sveta | 22360 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

10. Агрегация и GROUP BY-1 запрос.

Вывести все журналы и число их клиентов.

Select client, COUNT(DISTINCT title) FROM OrderClient GROUP BY client;

|  |  |
| --- | --- |
| **TITLE** | **COUNT(DISTINCTCLIENT)** |
| Teoreticheskaya i prikladnaya ekologiya | 3 |
| Functional Ecology | 1 |
| Vokrug sveta | 1 |
| Nature | 1 |
| Journal of Ecology | 2 |
| GEO | 1 |

11. Агрегация, GROUP BY, HAVING- 1 запрос.

Вывести журнал, имя клиента и его максимальный тираж журналов, которые появились раньше 1950 года.

Select Journal.title, name, max(edition) FROM Journal, WriteJornal where WriteJornal.title=Journal.title GROUP BY Journal.title, name HAVING MIN(year\_of\_creation)<1950;

|  |  |  |
| --- | --- | --- |
| **TITLE** | **NAME** | **MAX(EDITION)** |
| Yunyy naturalist | Newton Melissa | 20000 |
| Journal of Ecology | Charles Gibson | 65000 |
| Journal of Ecology | Oparin Maksim | 65000 |
| Journal of Ecology | Allison Cook | 65000 |
| Vokrug sveta | Ramsey Jack | 100000 |
| Journal of Ecology | Petrova Maria | 65000 |

12. Коррелированный запрос – 2 запроса разного типа.

Найти номера продукции журналов, печать которых пережила больше 400 изданий

Select DISTINCT production\_number FROM Journal WHERE Journal.title IN(Select Publishing\_house.title FROM Publishing\_house WHERE Journal.title=Publishing\_house.title) and number\_of\_issues>400;

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |
| --- |
| **PRODUCTION\_NUMBER** |
| 22360 |
| 23668 |
| 78023 |

Найти имена и специальность авторов, которые у которых нет клиентов.

Select name, specialization FROM Authors WHERE author\_number <>ANY(Select clients\_number FROM Clients where place\_of\_work=client);

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |
| --- | --- |
| **NAME** | **SPECIALIZATION** |
| Charles Gibson | Professor |
| Newton Melissa | Сandidate of Sciences |
| Oparin Maksim | Dean |

3. Триггеры базы данных

В стандарте SQL3 предусмотрены триггеры – обработчики, выполняющие проверку ограничений. Условия проверки ограничений задаются программно, в ответ на событие производится действие, которое может выполнятся как до наступления, так и вместо или после наступления события.

1. Триггер уровня строки, использующий псевдозаписи old предназначенный для поддержания ссылочной целостности.

При изменениях (удалении) данных в одной таблице триггер удаляет данные и в других таблицах, на которые есть ссылка.

create or replace trigger Journal\_trigger

after delete on Journal

for each row

begin

delete from Publishing\_house where title=:old.title;

delete from WriteJornal where title=:old.title;

delete from OrderClient where title=:old.title;

end Journal\_trigger;

Изначально в таблицах есть записи о журнале Journal of Ecology.

select distinct title from Journal NATURAL JOIN WriteJornal NATURAL JOIN OrderClient;

|  |
| --- |
| **TITLE** |
| Teoreticheskaya i prikladnaya ekologiya |
| Functional Ecology |
| Vokrug sveta |
| Journal of Ecology |
| GEO |
| Nature |

Удалим все записи, связанные с этим журналом.

delete from Journal where title = 'Journal of Ecology';

select distinct title from Journal NATURAL JOIN WriteJornal NATURAL JOIN OrderClient;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **TITLE** | | Teoreticheskaya i prikladnaya ekologiya | | Functional Ecology | | Vokrug sveta | | GEO | | Nature | |

2. Триггер с предикатами IF-THEN для наиболее полного аудита действий пользователя

Создадим новую таблицу аудита клиентов, в которую будем записывать все изменения данных (пользователь который вносил изменения (userID) , системное время совершения операции (time), операция (operation) и само изменение (history).

create table Client\_audit (userID varchar2 (30), time Timestamp, operation varchar2 (20), history varchar2(60));

Этот триггер позволяет проверять, какое действие вызвало его срабатывание и отслеживать изменения кортежей таблицы.

create or replace trigger Client\_trigger

after insert or delete or update on Clients

for each row

declare

making varchar2(30);

begin

if inserting then making :='INSERTING'; End if;

if deleting then making :='DELETING'; End if;

if updating then making :='UPDATING'; End if;

insert into Client\_audit values (user, sysdate, making ,

'oldName:'||:old.client||'newName:'||:new.client);

End Client\_trigger;

В для иллюстрации работы триггера добавим в таблицу клиентов запись о новом клиенте, затем переименуем его и удалим.

INSERT INTO Clients VALUES ('MIT', '14 Stoun Street', '3(204)565-02-88', 98201);

update Clients set client='Oxford' where client='MIT';

delete from Clients where client='Oxford';

Выведем данные в таблице аудита клиентов.

select \* from Client\_audit;

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **USERID** | **TIME** | **OPERATION** | **HISTORY** |
| ANONYMOUS | 23-DEC-18 09.34.59.000000 PM | INSERTING | oldName:newName:MIT |
| ANONYMOUS | 23-DEC-18 09.36.20.000000 PM | UPDATING | oldName:MITnewName:Oxford |
| ANONYMOUS | 23-DEC-18 09.36.33.000000 PM | DELETING | oldName:OxfordnewName: |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

3. Триггер с When

Отражает изменение данных в таблице по условию When.

Создадим таблицу популярных журналов, в которую будем записывать название журнала, год его создания и тираж.

create table Popular\_Jornal (title varchar2(100), year\_of\_creation int, edition int);

Для отслеживания вставки или обновления в таблицу журналов создадим триггер Popular\_Journal\_trigger, который будет фиксировать все действия с таблицей, в результате которых тираж будет выше 70000.

create or replace trigger Popular\_Journal\_trigger

after insert or update on Journal

for each row

when (new.edition>70000 or old.edition>70000)

begin

if inserting then

insert into Popular\_Journal

(title, year\_of\_creation, edition)

values (:new.title,:new.year\_of\_creation,:new.edition); end if;

if updating and :old.edition!=:new.edition then

insert into Popular\_Journal

(title, year\_of\_creation, edition)

values (:old.title,:old.year\_of\_creation,:new.edition); End if;

End Popular\_Journal\_trigger;

Для демонстрации работы триггера обновим тираж журнала GEO.

update Journal set edition=80000 where title='GEO';

И выведем содержимое таблицы популярных журналов.

select \* from Popular\_Journal;

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **TITLE** | **YEAR\_OF\_CREATION** | **EDITION** |
| GEO | 1976 | 80000 |

4. Программирование клиента базы данных

При разработке баз данных операторы языка SQL включаются в программу, составленную на основном языке программирования, в данном случае Java. Препроцессор заменяет операторы SQL на функции основного языка и может компилировать полученный код.

4.1 Первая транзакция

Позволяет в случае сбоя в системе, ошибки, возникновения исключения сохранить корректность вводимых данных за счет атомарности транзакции. Т.е. данные могут быть или внесены в БД полностью либо произойдет откат к предыдущему состоянию (до момента начала выполнения транзакции).

В данном случае транзакция изменяет название компании, при этом ей необходимо обновить все связанные с ней дочерние записи, поэтому в таблицах, где первичный кортеж-название компании не является внешним ключом идет обновление данных, а в связанных ссылочным ограничением происходит последовательное удаление и вставка соответствующих строк. В случае ошибки или неудачного перевода из какой-либо кортежа, транзакция будет полностью отменена.

public static void company\_renaming (Connection con) throws SQLException {

int clients\_number=13650;

String client=null, newclient="MIT";

ArrayList <String> title=new <String> ArrayList();

try {

con.setAutoCommit(false);

Statement st = con.createStatement();

ResultSet man=st.executeQuery("select client from Clients where clients\_number="+clients\_number+"");

if(man.next())

client = man.getString(1);

System.out.println(client+"\t"+clients\_number);

man=st.executeQuery("select title from OrderClient where client='"+client+"'");

while(man.next())

title.add(man.getString(1));

st.executeUpdate("delete from OrderClient where client='"+client+"'");

st.executeUpdate("update Authors set place\_of\_work='"+newclient+"' where place\_of\_work='"+client+"'");

st.executeUpdate("update Clients set client='"+newclient+"' where clients\_number="+clients\_number+"");

for(int i=0; i<title.size();i++) {

st.executeUpdate("insert into OrderClient values ('"+newclient+"', '"+title.get(i)+"')");

}

man=st.executeQuery("select client from Clients where clients\_number="+clients\_number+"");

if(man.next())

client = man.getString(1);

System.out.println(client+"\t"+clients\_number);

con.commit();

} catch (SQLException e) {

e.printStackTrace();

con.rollback();

}

}

4.2 Вторая транзакция

Для обеспечения изоляции параллельно выполняемых операций был выбран режим изоляции транзакций READ COMMITED, который устанавливается с помощью функции setTransactionIsolation() и позволяет запретить случаи потерянного обновления и грязного чтения данных.

В приведенном коде производится изменение номера телефона издательства, сначала пользователь вводит номер и после обращения к таблице издательства его обновляет, за счет режима изоляции READ COMMITED исключается ситуация, когда второй пользователь параллельно обращается к издательству по старому номеру телефона.

public static void change\_of\_number(Connection con) throws SQLException {

boolean ans=true;

String name=null, number="33(711)655-11-99";

int index=0;

try{

con.setTransactionIsolation(2);

con.setAutoCommit(false);

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select name\_publ, index2 from Journal,Publishing\_house where Journal.title=Publishing\_house.title and phone\_number='"+number+"'");

if (res.next()) {

name = res.getString(1);

index=res.getInt(2);

}

System.out.println(name+"\t "+index+"\t "+number);

ans=false;

if(!ans)

{

number="00(45)512-11-03";

PreparedStatement st1=con.prepareStatement("update Publishing\_house set phone\_number=? where index2=?");

st1.setString(1, number);

st1.setInt(2, index);

int val=st1.executeUpdate();

}

con.commit();

res=st.executeQuery("select name\_publ, index2 from Journal,Publishing\_house where Journal.title=Publishing\_house.title and phone\_number='"+number+"'");

if (res.next()) {

name = res.getString(1);

index=res.getInt(2);

}

System.out.println(name+"\t "+index+"\t "+number);

}catch(SQLException e) {

con.rollback();

con.close(); }

}

Заключение

В ходе написания данной курсовой работы были изучены специальные источники по выбранной тематике, также были приобретены ключевые навыки работы с СУБД Oracle 11 g Express Edition, проектирования моделей баз данных с помощью моделей «сущность-связь» и языка определения данных ODL, был получен необходимый опыт создание таблиц на языке SQL и написание к ним запросов. Также попутно были изучены основные приемы программирования базы данных с помощью языка PL\SQL, такие как курсоры и треггеры, и методов JDBS.

Список источников

1. Малыхина Г.Ф. Управление данными: учеб. пособие. – СПб.: Изд-во Политехн. ун-та, 2012.252 с.
2. Нокс Д. Oracle Database 10g. Создание эффективной системы безопасности. – М.:Лори, 2007.-556с.
3. Лафоре Р. Структуры данных и алгоритмы в Java. Классика Computers Science. 2-е изд. - СПб.: Питер, 2013. — 704 с.:
4. <https://docs.oracle.com>
5. <https://stackoverflow.com/questions>
6. <https://habr.com>
7. <http://www.sql.ru/forum/>