

Санкт-Петербургский Государственный Политехнический Университет
Факультет Технической Кибернетики
Кафедра компьютерных систем и программных технологий

ОТЧЕТ
о лабораторной работе

Тема: разработка и проектирование базы данных и консольного приложения

Выполнил студент:

гр. 4081/11
Дорогов А. И

Преподаватель:

Цесько В. А

Санкт-Петербург
2011 г.

Проектирование БД и разработка ER диаграммы

Задание: CRM-система

Содержит информацию о компаниях, контактах, встречах, звонках, мероприятиях, рекламных кампаниях, документах, сделках и пр. По каждой компании/контакту ведется история звонков, встреч, e-mail-писем.

План работ:

- Спроектировать БД
- Построить ER диаграмму
- Привести БД к 3НФ
- Создать скрипт создания необходимых таблиц
- Создать скрипт реализующий sql запросы
- Наполнение БД тестовыми данными

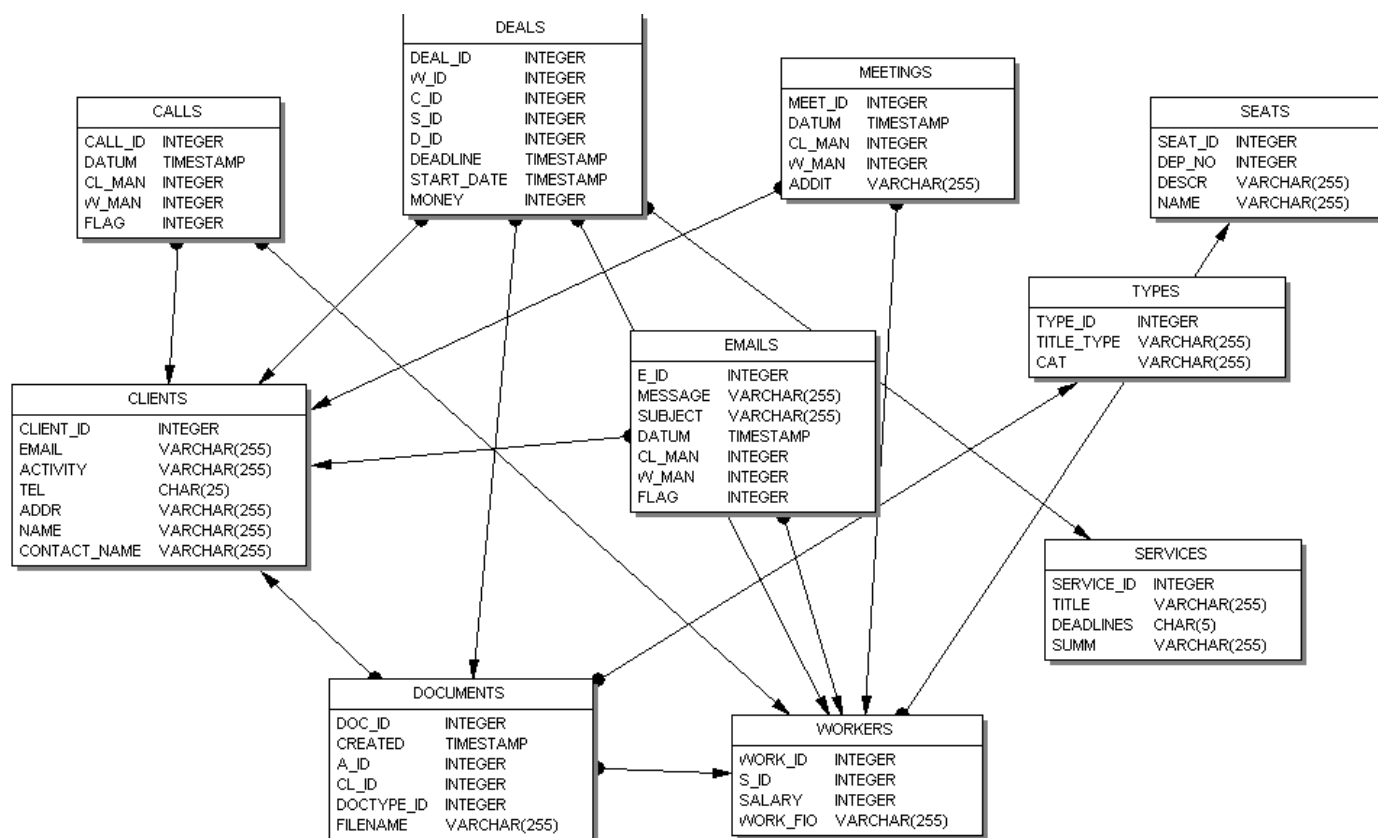


Рис. ER диаграмма спроектированной базы данных

Каждая таблица содержит осмысленное название поэтому не нужды описывать каждую из таблиц в отдельности.

База данных изначально проектировалась по возможности в нормальной форме, и в данном случае по заданию она соответствует форме 3.

Скрипт создания таблиц существует в svn в ревизии 83. create_tables.sql

```
create table seats(
    seat_id int not null,
    dep_no int not null,
    descr varchar(255),
    name varchar(255),
    primary key(seat_id)
);

create table services(
    service_id int not null,
    title varchar(255),
    deadlines char(5),
    summ varchar(255),
    primary key(service_id)
);

create table types(
    type_id int not null,
    title_type varchar(255) not null,
    cat varchar(255) not null,
    primary key(type_id)
);

create table clients(
    client_id int not null,
    email varchar(255) unique,
    activity varchar(255),
    tel char(25),
    addr varchar(255),
    name varchar(255),
    contact_name varchar(255),
    primary key (client_id)
);

create table workers(
    work_id int not null,
    s_id int not null,
    salary int not null,
    work_fio varchar(255) not null,
    primary key (work_id),
    foreign key (s_id) references seats(seat_id)
);

create table documents(
    doc_id int not null,
    created timestamp not null,
    a_id int not null,
    cl_id int not null,
    doctype_id int not null,
    filename varchar(255) not null,
    primary key (doc_id),
    foreign key (doctype_id) references types(type_id),
    foreign key (cl_id) references clients(client_id),
```

```

        foreign key (a_id) references workers(work_id)
    );

create table deals(
    deal_id int not null,
    w_id int not null,
    c_id int not null,
    s_id int not null,
    d_id int not null,
    deadline timestamp,
    start_date timestamp,
    money int not null,
    primary key (deal_id),
    foreign key (c_id) references clients(client_id),
    foreign key (w_id) references workers(work_id),
    foreign key (d_id) references documents(doc_id),
    foreign key (s_id) references services(service_id)
);

create table emails(
    e_id int not null,
    message varchar(255),
    subject varchar(255),
    datum timestamp,
    cl_man int not null,
    w_man int not null,
    flag int not null,
    primary key(e_id),
    foreign key(cl_man) references clients(client_id),
    foreign key(w_man) references workers(work_id)
);

create table calls(
    call_id int not null,
    datum timestamp,
    cl_man int not null,
    w_man int not null,
    flag int not null,
    primary key(call_id),
    foreign key(cl_man) references clients(client_id),
    foreign key(w_man) references workers(work_id)
);

create table meetings(
    meet_id int not null,
    datum timestamp,
    cl_man int not null,
    w_man int not null,
    addit varchar(255),
    primary key(meet_id),
    foreign key(cl_man) references clients(client_id),
    foreign key(w_man) references workers(work_id)
);

```

Скрипт с примером добавления файлов в БД приведен в svn в ревизии 84 **inserts.sql**.

```

connect 'tiger.ftk.spbstu.ru:/var/lib/firebird/4081_2/crm.fdb' user 'SYSDBA'
password 'masterkey';

```

```

/* seats */
insert into seats values('0', '1', 'Description seat','Coder');
insert into seats values('1', '2', 'Description seat','Manager');
insert into seats values('2', '3', 'Description seat','Director');
insert into seats values('3', '4', 'Description seat','Driver');
insert into seats values('4', '5', 'Description seat','Controller');

/* services */
insert into services values('0', 'Web-project', '30','50000');
insert into services values('1', 'Design', '20','30000');
insert into services values('2', 'Android app', '10','10000');
insert into services values('3', 'Iphone App', '20','15000');
insert into services values('4', 'Symbian App', '20','5000');

/* types */
insert into types values('0', 'Contract', '/cont/');
insert into types values('1', 'Contracct_additional', '/contadd/');
insert into types values('2', 'TZ','/tzs/');
insert into types values('3', 'Fixes', '/fixes/');
insert into types values('4', 'Others', '/others/');

/* CLients */
insert into clients values('0', 'email_0@gmail.com', 'Activity 0', '3844', 'spb 5248',
'Company 0', 'CName 0');
insert into clients values('1', 'email_1@gmail.com', 'Activity 1', '3681', 'spb 4616',
'Company 1', 'CName 1');
insert into clients values('2', 'email_2@gmail.com', 'Activity 2', '3833', 'spb 4489',
'Company 2', 'CName 2');
insert into clients values('3', 'email_3@gmail.com', 'Activity 3', '4043', 'spb 856',
'Company 3', 'CName 3');
insert into clients values('4', 'email_4@gmail.com', 'Activity 4', '3524', 'spb 5237',
'Company 4', 'CName 4');
insert into clients values('5', 'email_5@gmail.com', 'Activity 5', '5511', 'spb 2435',
'Company 5', 'CName 5');
insert into clients values('6', 'email_6@gmail.com', 'Activity 6', '1196', 'spb 1824',
'Company 6', 'CName 6');
insert into clients values('7', 'email_7@gmail.com', 'Activity 7', '4349', 'spb 4234',
'Company 7', 'CName 7');
insert into clients values('8', 'email_8@gmail.com', 'Activity 8', '3364', 'spb 4411',
'Company 8', 'CName 8');
insert into clients values('9', 'email_9@gmail.com', 'Activity 9', '2706', 'spb 1891',
'Company 9', 'CName 9');
insert into clients values('10', 'email_10@gmail.com', 'Activity 10', '1910', 'spb
1196', 'Company 10', 'CName 10');
insert into clients values('11', 'email_11@gmail.com', 'Activity 11', '4879', 'spb
2195', 'Company 11', 'CName 11');
insert into clients values('12', 'email_12@gmail.com', 'Activity 12', '3649', 'spb
3708', 'Company 12', 'CName 12');
insert into clients values('13', 'email_13@gmail.com', 'Activity 13', '4104', 'spb
695', 'Company 13', 'CName 13');
insert into clients values('14', 'email_14@gmail.com', 'Activity 14', '1374', 'spb
4263', 'Company 14', 'CName 14');
insert into clients values('15', 'email_15@gmail.com', 'Activity 15', '3853', 'spb
1297', 'Company 15', 'CName 15');
insert into clients values('16', 'email_16@gmail.com', 'Activity 16', '3135', 'spb
2207', 'Company 16', 'CName 16');
insert into clients values('17', 'email_17@gmail.com', 'Activity 17', '977', 'spb
2661', 'Company 17', 'CName 17');
insert into clients values('18', 'email_18@gmail.com', 'Activity 18', '1391', 'spb
4478', 'Company 18', 'CName 18');
insert into clients values('19', 'email_19@gmail.com', 'Activity 19', '5473', 'spb
1592', 'Company 19', 'CName 19');

```

```

/*workers */
insert into workers values('0', '1', '15865', 'Name0 Lastname0');
insert into workers values('1', '4', '8727', 'Name1 Lastname1');
insert into workers values('2', '0', '2468', 'Name2 Lastname2');
insert into workers values('3', '3', '1455', 'Name3 Lastname3');
insert into workers values('4', '4', '5924', 'Name4 Lastname4');
insert into workers values('5', '1', '8285', 'Name5 Lastname5');
insert into workers values('6', '0', '19394', 'Name6 Lastname6');
insert into workers values('7', '1', '3505', 'Name7 Lastname7');
insert into workers values('8', '4', '19040', 'Name8 Lastname8');
insert into workers values('9', '3', '9448', 'Name9 Lastname9');
insert into workers values('10', '0', '15026', 'Name10 Lastname10');
insert into workers values('11', '4', '14263', 'Name11 Lastname11');
insert into workers values('12', '3', '7436', 'Name12 Lastname12');
insert into workers values('13', '3', '19143', 'Name13 Lastname13');
insert into workers values('14', '3', '16502', 'Name14 Lastname14');
insert into workers values('15', '0', '1851', 'Name15 Lastname15');
insert into workers values('16', '1', '15869', 'Name16 Lastname16');
insert into workers values('17', '3', '6125', 'Name17 Lastname17');
insert into workers values('18', '2', '387', 'Name18 Lastname18');
insert into workers values('19', '1', '18838', 'Name19 Lastname19');

/* DDocuments */
insert into documents values('0', '2011-03-18:12-54-38',
'13','6','1','Filename_0.pdf');
insert into documents values('1', '2011-03-18:12-54-38',
'7','10','2','Filename_1.pdf');
insert into documents values('2', '2011-03-18:12-54-38',
'17','9','0','Filename_2.pdf');
insert into documents values('3', '2011-03-18:12-54-38',
'13','15','3','Filename_3.pdf');
insert into documents values('4', '2011-03-18:12-54-38', '9','0','3','Filename_4.pdf');
insert into documents values('5', '2011-03-18:12-54-38',
'18','11','3','Filename_5.pdf');
insert into documents values('6', '2011-03-18:12-54-38', '8','3','4','Filename_6.pdf');
insert into documents values('7', '2011-03-18:12-54-38',
'15','16','4','Filename_7.pdf');
insert into documents values('8', '2011-03-18:12-54-38',
'19','3','1','Filename_8.pdf');
insert into documents values('9', '2011-03-18:12-54-38',
'14','18','1','Filename_9.pdf');
insert into documents values('10', '2011-03-18:12-54-38',
'0','11','1','Filename_10.pdf');
insert into documents values('11', '2011-03-18:12-54-38',
'19','17','2','Filename_11.pdf');
insert into documents values('12', '2011-03-18:12-54-38',
'9','17','3','Filename_12.pdf');
insert into documents values('13', '2011-03-18:12-54-38',
'11','4','0','Filename_13.pdf');
insert into documents values('14', '2011-03-18:12-54-38',
'16','3','3','Filename_14.pdf');
insert into documents values('15', '2011-03-18:12-54-38',
'6','3','1','Filename_15.pdf');
insert into documents values('16', '2011-03-18:12-54-38',
'0','7','1','Filename_16.pdf');
insert into documents values('17', '2011-03-18:12-54-38',
'7','9','3','Filename_17.pdf');
insert into documents values('18', '2011-03-18:12-54-38',
'19','2','1','Filename_18.pdf');
insert into documents values('19', '2011-03-18:12-54-38',
'6','0','4','Filename_19.pdf');
/* deals */

```

```

insert into deals values('0', '17', '3', '4','0','2011-03-18:12-48-58','2011-03-18:12-48-58','58246');
insert into deals values('1', '6', '15', '4','1','2011-03-18:12-48-58','2011-03-18:12-48-58','16489');
insert into deals values('2', '6', '13', '2','2','2011-03-18:12-48-58','2011-03-18:12-48-58','44849');
insert into deals values('3', '17', '10', '4','3','2011-03-18:12-48-58','2011-03-18:12-48-58','90723');
insert into deals values('4', '15', '3', '0','4','2011-03-18:12-48-58','2011-03-18:12-48-58','95879');
insert into deals values('5', '10', '16', '1','5','2011-03-18:12-48-58','2011-03-18:12-48-58','69995');
insert into deals values('6', '5', '3', '0','6','2011-06-18:12-48-58','2011-03-18:12-48-58','22144');
insert into deals values('7', '7', '9', '4','7','2011-05-18:12-48-58','2011-03-18:12-48-58','81378');
insert into deals values('8', '16', '2', '2','8','2011-03-18:12-48-58','2011-03-18:12-48-58','30550');
insert into deals values('9', '6', '0', '1','9','2011-03-18:12-48-58','2011-03-18:12-48-58','67083');
insert into deals values('10', '15', '8', '2','10','2011-06-18:12-48-58','2011-03-18:12-48-58','41251');
insert into deals values('11', '7', '7', '3','11','2011-03-18:12-48-58','2011-03-18:12-48-58','8427');
insert into deals values('12', '18', '17', '3','12','2011-03-18:12-48-58','2011-03-18:12-48-58','14950');
insert into deals values('13', '10', '1', '2','13','2011-07-18:12-48-58','2011-03-18:12-48-58','93128');
insert into deals values('14', '13', '16', '4','14','2011-03-18:12-48-58','2011-03-18:12-48-58','96958');
insert into deals values('15', '4', '5', '2','15','2011-03-18:12-48-58','2011-03-18:12-48-58','31275');
insert into deals values('16', '7', '18', '3','16','2011-05-18:12-48-58','2011-03-18:12-48-58','89068');
insert into deals values('17', '1', '0', '1','17','2011-03-18:12-48-58','2011-03-18:12-48-58','71581');
insert into deals values('18', '14', '3', '4','18','2011-04-18:12-48-58','2011-03-18:12-48-58','93525');
insert into deals values('19', '16', '7', '0','19','2011-03-18:12-48-58','2011-03-18:12-48-58','13516');
/* Emails */
insert into emails values('0', 'Message_69', 'Subject_77','2011-03-18:12-58-05','5','13','1');
insert into emails values('1', 'Message_198', 'Subject_60','2011-03-18:12-58-05','18','15','1');
insert into emails values('2', 'Message_82', 'Subject_26','2011-03-18:12-58-05','6','12','0');
insert into emails values('3', 'Message_80', 'Subject_8','2011-03-18:12-58-05','10','15','0');
insert into emails values('4', 'Message_131', 'Subject_57','2011-03-18:12-58-05','11','5','1');
insert into emails values('5', 'Message_186', 'Subject_38','2011-03-18:12-58-05','9','12','0');
insert into emails values('6', 'Message_85', 'Subject_81','2011-03-18:12-58-05','2','14','1');
insert into emails values('7', 'Message_22', 'Subject_26','2011-03-18:12-58-05','7','16','1');
insert into emails values('8', 'Message_125', 'Subject_89','2011-03-18:12-58-05','12','16','1');
insert into emails values('9', 'Message_16', 'Subject_88','2011-03-18:12-58-05','14','18','1');

```

```

insert into emails values('10', 'Message_72', 'Subject_10','2011-03-18:12-58-
05','9','6','0');
insert into emails values('11', 'Message_125', 'Subject_21','2011-03-18:12-58-
05','17','8','1');
insert into emails values('12', 'Message_126', 'Subject_45','2011-03-18:12-58-
05','15','8','1');
insert into emails values('13', 'Message_117', 'Subject_30','2011-03-18:12-58-
05','12','12','1');
insert into emails values('14', 'Message_55', 'Subject_10','2011-03-18:12-58-
05','18','0','1');
insert into emails values('15', 'Message_44', 'Subject_41','2011-03-18:12-58-
05','2','18','1');
insert into emails values('16', 'Message_160', 'Subject_67','2011-03-18:12-58-
05','13','5','0');
insert into emails values('17', 'Message_198', 'Subject_73','2011-03-18:12-58-
05','8','1','0');
insert into emails values('18', 'Message_151', 'Subject_4','2011-03-18:12-58-
05','18','5','0');
insert into emails values('19', 'Message_16', 'Subject_9','2011-03-18:12-58-
05','13','14','1');
commit;

```

Необходимо отметить наличие средств для автоматического заполнения БД тестовыми данными к таким средствам относятся например IVExpert, приведем скриншот диалога заполнения БД тестовыми данными.

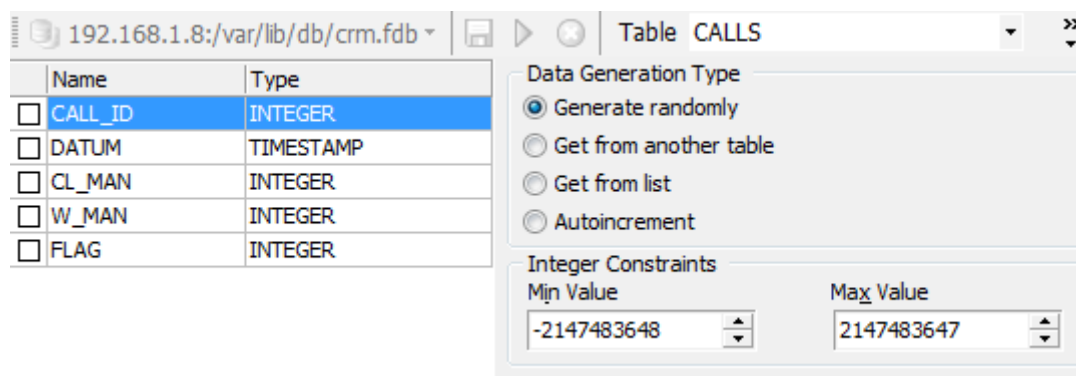


Рис. Заполнение БД тестовыми данными в IVExpert

Так же в качестве генератора тестовых данных был испробован FlameRobin который так же предлагает удобные средства для работы БД firebird.

Выводы: задания этой главы позволяют познакомиться с процессом проектирования базы данных, создания скриптов для добавления необходимых таблиц в базу данных. Эти скрипты могут использоваться в дальнейшем для автоматического создания базы данных какого-либо приложения. Так же были исследованы средства для работы с firebird это приложений isql, которое существует как под windows так и под linux. Так же была исследована работы визуальных средств работы с БД это ibexpert который имеет ряд полезных функций для работы, в частности генератор тестовых данных.

Индивидуальные запросы

Индивидуальные запросы:

1. Выбрать все документы принадлежащие клиенту и работнику которые указаны в письме.
2. Отменить встречу если работник и клиент не созванивались 2 месяца.
3. Вывести все сделки 3-х наиболее активно общающихся клиентов и работников.

Запросы:

1)

```
select * from documents WHERE  
cl_id=(select cl_man FROM emails WHERE e_id=3)  
and  
a_id = (select w_man FROM emails WHERE e_id=3);
```

2) в данном случае это хранимая процедура

```
SET TERM ^ ;  
create or alter procedure NEW_PROCEDURE  
as  
declare variable M_ID integer;  
begin  
for select meet_id from meetings WHERE meet_id IN  
(select m.MEET_ID from meetings m, calls c where datediff(day from c.DATUM to  
current_timestamp)<50  
and m.W_MAN=c.W_MAN and c.CL_MAN=m.CL_MAN)  
into :M_ID  
do  
DELETE from MEETINGS WHERE meet_id = :M_ID;  
end^  
SET TERM ; ^  
GRANT SELECT,DELETE ON MEETINGS TO PROCEDURE NEW_PROCEDURE;  
GRANT SELECT ON CALLS TO PROCEDURE NEW_PROCEDURE;  
GRANT EXECUTE ON PROCEDURE NEW_PROCEDURE TO SYSDBA;
```

3) здесь обычный вложенный запрос

```
select * from DEALS where w_id in  
(select w_man from (select first 3 w_man, cl_man, count(*) as cnt from CALLS  
group by w_man, cl_man order by cnt desc)) and c_id in  
(select cl_man from (select first 3 w_man, cl_man, count(*) as cnt from CALLS  
group by w_man, cl_man order by cnt desc))
```

Выводы: в результате выполнения данного пункта были изучены вопросы связанные с реализацией запросов к БД, с использованием вложенных запросов и другие средства. Необходимо отметить создание ХП в данном случае запрос выполнен в виде ХП так как довольно часто такие модификации данных в БД (удаление, обновление) часты и их удобнее реализовывать в виде ХП для последующего обращения.

Транзакции firebird 2.5

1. Исследование уровня транзакций Snapshot

T1 – Tn – транзакции

T1: уровень Snapshot

```
select * from types where type_id=0
```

T2: уровень Read Committed

```
update types set title_type='Contract' where type_id=0
```

Запустим на выполнение T1, и T2 и подтверждая T2, получим:

TYPE_ID	TITLE_TYPE
0	New title

Это показывает то что T1 использует старую информацию, ее слепок до выполнения всех транзакций. Вернем значение обратно в New title.

2. Исследование Read Committed

T1: read committed

T2: rw stability

```
update types set title_type='Contract' where type_id=0
```

Запускаем T2 не подтверждая, получаем результаты первой транзакции:

TYPE_ID	TITLE_TYPE
0	New title

Попробуем записать первой транзакцией что-нибудь в эту таблицу.

Messages

Unsuccessful execution caused by system error that does not preclude successful execution of subsequent statements.
lock conflict on no wait transaction.

Это говорит о том что доступ к этой таблице возможен только для чтения, а на запись она заблокирована транзакцией T2 которая стартовала как RW stability.

Теперь подтвердим T2. Получим результат первой транзакции:

TYPE_ID	TITLE_TYPE
0	Contract

Как видно данные обновились, тогда как в случае с Snapshot нам показывалась старая информация, можно сказать что read committed показывает все подтвержденные

транзакции, а снимок - соответственно показывает данные которые были подтверждены до запуска транзакции.

3. RW Table Stability

Данные уровни существуют для предотвращения одновременного доступа к ресурсу, например одной и той же записи в таблице. Если запустить транзакцию в режиме **RW stability** то это приведет к тому, что другие транзакции не смогут получить доступ к данной таблице для записи, а только для чтения. Продемонстрируем:

T1: RW stability

```
update types set title_type='Contract' where type_id=0
```

T2: Read committed

```
update types set title_type='Others 5' where type_id=4
```

После выполнения T2 и не завершения T1 получим:

Messages |

Unsuccessful execution caused by system error that does not preclude successful execution of subsequent statements.
lock conflict on no wait transaction.

T2:

```
select first 2 * from types
```

TYPE_ID	TITLE_TYPE
0	Contract_5
1	Contracct_additional

Как видно, доступ на чтение есть.

T2:

```
update types set title_type='Others 5' where type_id=4
```

Завершим T1 и запустим T2, получим:

TYPE_ID	TITLE_TYPE
0	Contract_5
1	Contracct_additional
2	TZ
3	Fixes
4	Others 5

Как видно данные обновились, тоесть T2 получило доступ к таблице после окончания транзакции T1.

T1: RW stability

```
select * from types where type_id=0
```

T2: RO stability

```
select first 2 * from types
```

TYPE_ID	TITLE_TYPE
0	Contract_5
1	Contract_additional

Таким образом операции выборки из БД не влияют на доступ к таблицам в режиме RW stability.

4. RO Table Stability

T1: RO stability

```
update types set title_type='Contract_8' where type_id=0
```

Результат:

Messages |

The INSERT, UPDATE, DELETE, DDL or authorization statement cannot be executed because the transaction is inquiry only. attempted update during read-only transaction.

Что говорим о том, что это должна быть операция чтения.

T2: RW stability

```
update types set title_type='Contract_5' where type_id=0
```

Messages |

Unsuccessful execution caused by system error that does not preclude successful execution of subsequent statements. lock conflict on no wait transaction.

T2: Read committed

Messages |

Unsuccessful execution caused by system error that does not preclude successful execution of subsequent statements. lock conflict on no wait transaction.

Собственно до закрытия транзакции T1 нет возможности ни писать, ни читать из базы.

T1: RO Stability

```
select first 2 * from types
```

T2: RO stability, RW stability

```
select * from types where type_id=0
```

Данные варианты возможны.

T2: RW Stability, Read committed, Snapshot

```
update types set title_type='Contract_8' where type_id=0
```

Это невозможно.

Messages |

Unsuccessful execution caused by system error that does not preclude successful execution of subsequent statements. lock conflict on no wait transaction.

Выводы:

В работе исследовались различные уровни изоляций для firebird 2.5. В качестве вывода можно сформулировать некоторые правила по использованию различных уровней изоляции.

1. Когда требуются актуальные данные, необходимо использовать Read committed. Но данный тип транзакции не обеспечивает повторимости чтения, так как подтвержденные данные становятся видны этой транзакции.
2. Когда необходимо предотвратить изменение одних и тех же данных, можно выставить на эту транзакцию уровень изоляции RW stability, это позволит запретить запись в таблицу других процессов, но оставит чтение.
3. Слепок может обеспечить повторимость чтения, но обладает не самой актуальной информацией, так как создает список транзакций завершенных до ее старта.

Разработка триггеров

1. Триггер который заполняет ключевое поле

В отличие от mysql в firebird отсутствует возможность использования автоинкремента «из коробки» для этого можно создать генератор или последовательность и создать триггеры который будет высчитывать новый идентификатор перед записью в таблицу.

```
CREATE SEQUENCE GENERATORID;  
ALTER SEQUENCE GENERATORID RESTART WITH 0;  
  
// создаем триггера для автоинкремента  
CREATE OR ALTER TRIGGER TYPES_AUTOINCREMENT FOR TYPES  
ACTIVE BEFORE INSERT POSITION 0  
as  
begin  
    if ((new.TYPE_ID is null) or (new.TYPE_ID = 0)) then  
        begin  
            new.TYPE_ID = gen_id(GENERATORID, 1);  
        end  
    end  
end  
^  
SET TERM ; ^
```

Пример использования

```
insert into types (title_type, cat) values('Smth', '/category')
```

Здесь триггер самостоятельно рассчитает идентификатор для новой записи.

Результат:

TYPE_ID	TITLE_TYPE
0	Contract_5
1	Contracct_additional
2	TZ
3	Fixes
4	Others 5
5	Smth

2. Триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице

При удалении документа происходит нарушение целостности, так как запись в таблице сделок становится бессмысленной если не будет существовать документа с идентификатором который указан в этой записи, для этого создадим 2 триггера на удаление и на обновление данных. Если удаляется документ – удаляем сделку, если у документа меняется идентификатор – меняем идентификатор документа в таблицу сделок.

Пример:

Выключим триггер, попробуем удалить запись о документе.

9	14.05.2000 08:04	85	31	3	lAgcSBenWIpPVzFimUPuGUjXJcVZVALhbSJtnhqWwFRVL
10	29.09.1998 06:57	48	73	0	iEwIMjexZvbRKOPnpjCYZLJoPjECGKnPoKabTfzSBAKlpAz
11	19.05.1980 04:01	97	45	3	pXnLnMDpflZJjwprFrRpyzqHRpnpFWQUteGgRjwXUvGe
12	21.05.2002 14:57	82	34	0	cFLAzjDPLFTrGydPvMDhvaczRGQYHtbnfBNofDaLwrsUi
13	23.11.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
14	26.09.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
15	07.09.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
16	10.09.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
17	13.09.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
18	19.09.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
19	14.09.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
20	03.09.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
21	02.09.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
22	19.09.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
23	21.09.1999 18:20	82	71	1	CAqMoCjqWeULKAFbjxZOrmCftMtMYYoypaMSkdOpXze
24	18.09.1988 15:02	31	53	4	AoxzSkgTOPbBGbaMeyzoMsGtWmKSeMssjSkuYEjOFKu

Error

violation of FOREIGN KEY constraint "".

violation of FOREIGN KEY constraint "INTEG_38" on table "DEALS".

Foreign key references are present for the record.

OK

Включим триггер:

До удаления документа.

```
select * from deals where d_id='10'
```

DEAL_ID	W_ID	C_ID	S_ID	D_ID	DEADLINE	START_DATE	MONEY
14	17	24	3	10	30.05.2000 17:19	12.12.1999 05:14	127 561
31	52	72	0	10	06.10.1992 21:47	14.03.1998 11:07	657 991

После удаления документа и подтверждения транзакции.

DEAL_ID	W_ID	C_ID	S_ID	D_ID	DEADLINE	START_DATE	MONEY
<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>

Как видим сделки удалились.

Продемонстрируем изменение о идентификаторе документа.

```
select * from deals where d_id='6'
```

DEAL_ID	W_ID	C_ID	S_ID	D_ID	DEADLINE	START_DATE	MONEY
18	87	65	3	6	<null>	04.03.2004 21:17	35 155
184	44	82	2	6	<null>	<null>	99 267

Изменим значение документа на 666666, и проверим эти записи в таблице сделок.

```
select * from deals where deal_id='18'
```

DEAL_ID	W_ID	C_ID	S_ID	D_ID	DEADLINE	START_DATE	MONEY
18	87	65	3	666 666	<null>	04.03.2004 21:17	35 155

Как видно триггера работает верно.

Разработка индивидуальной хранимой процедуры

Желаем расформировать какой-либо департамент, и все встречи которые назначены сотрудникам из этого департамента переназначить на какого-то одного сотрудника.

```
SET TERM ^ ;
create or alter procedure NEW_PROCEDURE (
    WORKER integer,
    SEAT integer)
as
declare variable WORKERS_IDS integer;
begin
    for select WORK_ID
    from WORKERS
    where S_ID = :SEAT
    into :WORKERS_IDS
    do
        update meetings set W_MAN = :worker where W_MAN= :workers_ids;
end^

SET TERM ; ^
GRANT SELECT ON WORKERS TO PROCEDURE NEW_PROCEDURE;
GRANT SELECT,UPDATE ON MEETINGS TO PROCEDURE NEW_PROCEDURE;
GRANT EXECUTE ON PROCEDURE NEW_PROCEDURE TO SYSDBA;
```

Соберемся расформировать например 4 департамент а встречи назначим на 10 работника. Проверим на 3х сотрудниках из 4 департамента, это сотрудники 8, 18, 19

Record: 21					21 records		
MEET_ID	DATUM	CL_MAN	W_MAN	ADDIT			
120	<null>	63	19	<null>			
561	24.10.1981 21:35	37	8	<null>			
869	06.03.1984 08:58	7	8	<null>			
227	27.11.1984 10:16	23	19	<null>			
292	08.12.1984 12:07	79	8	<null>			
372	22.04.1986 10:42	55	8	<null>			
405	22.07.1988 02:21	32	19	iddKevanuOwnslWaMHBVzLuvcaHtkrbsueCyADLUrhKueKglmdYwGbWNuHfwAzAdYaGKVYTcJngttSGvdESKFPyaxDwXdXabYGLTEFWoSChmVoIZSA			
686	08.09.1991 23:04	62	19	nLmuNCdqpWCLDtPJMHqvJSZHQuqEgEptZdWcuasrDdUXTDJaUfkMeRCzftdmxtFwxBySBRjfuDdcNHLNBQXoVPQuUJyggSLDpmbhOTrTNUVBChpEyr			
762	28.12.1992 04:54	15	8	qcZVghdOIXhMjtBelqxYLjDluBtRfnhIndYuhNdeKkrUEtYPJvOvFsGZTaRznZhBcgVjUyoEjgyOzxEJTtEyLkyFkpeymartWBnVPreVpTvNyEGRifCDIETi			
424	20.10.1995 09:54	88	19	<null>			

Получили 21 запись.

Запустим процедуру, в результате ее выполнения в встречах с идентификаторами например 561, 869, 227 проверим изменение w_man.

```
execute procedure new_procedure(10, 4)
```



```
select * from meetings where w_man in (8, 18, 19) order by datum
```

MEET_ID	DATUM	CL_MAN	W_MAN	ADDIT
<null>	<null>	<null>	<null>	<null>

Проверим встречи с номерами.

MEET_ID	DATUM	CL_MAN	W_MAN	ADDIT
227	27.11.1984 10:16	23	10	<null>
561	24.10.1981 21:35	37	10	<null>
869	06.03.1984 08:58	7	10	<null>

Как видно теперь ответственный за эту встречу назначен человек который был указан в параметрах процедуры.

Выводы:

В результате выполнения заданий по разработке хранимых процедур можно сказать следующее: хранимые процедуры удобны для частого использования, достаточно один раз создать процедуру, что помогает экономить время. Хранимые процедуры и язык psql предоставляет довольно гибкие средства для реализации сложных процедур.

Необходимо отметить что в хранимых процедурах возможно не только изменение данных в БД, но так же проводить различные вычисления.

Разработка консольного приложения

Необходимо разработать консольное приложение со следующими функциями:

1. Импорт встреч в БД из xml файла
2. Экспорт встреч в указанном диапазоне в xml файл

Программа запускается с 2 параметрами, -i filename для импорта данных в БД, -e filename для экспорта данных из БД.

Существует файл конфигурации, его состав:

```
jdbc:firebirdsql://192.168.1.8//var/lib/db/crm.fdb
sysdba
masterkey
org.firebirdsql.jdbc.FBDriver
15.05.2001 20:37:47
20.05.2005 20:37:47
```

Адрес БД, пользователь, пароль, какой драйвер использовать, дата начал периода и дата конца периода.

Таким образом программу можно использовать для разных типов БД.

В программе присутствует проверка при импорте на наличие такой встречи в БД, и если такая встреча есть, запись не импортируется. Ошибки связанные с неправильным форматом данных ловятся в исключениях.

Так же в программе имеется диалог с пользователем во время экспорта данных, если файл уже существует, то пользователю будет предложено ввести новое имя для файла экспорта или перезаписать существующий.

```
File D:/export.xml already exist, you want to overwrite them?  
Type yes to overwrite, or type new filename: yes|
```

Пример неверно заданного значения даты в файле конфигурации и результат работы приложения.

```
File D:/export.xml already exist, you want to overwrite them?  
Type yes to overwrite, or type new filename: yes  
SQL Error: ошибка входных данных GDS Exception. 335544334. conversion error from string "00"
```

Пример xml файла для импорта:

```
<meetings>  
<meet>  
  <date>2003-07-28 13:17:29</date>  
  <worker>20</worker>  
  <client>42</client>  
  <addit>some info</addit>  
</meet>  
<meet>  
  <date>2013-07-28 13:17:29</date>  
  <worker>21</worker>  
  <client>22</client>  
  <addit>some info</addit>  
</meet>  
.....  
</meetings>
```

Результат работы программы на экспорт:

```
<meetings>  
<meet>  
  <data>2003-04-29 08:00:49.0</data>  
  <client>45</client>  
  <worker>5</worker>  
  <addit>null</addit>  
</meet>  
<meet>  
  <data>2003-07-28 13:17:29.0</data>  
  <client>63</client>  
  <worker>78</worker>  
  <addit>thIJpdzJwtZdBLLwcGUIIVPjQrYxINwcVfLkikUFETIgFtCIawqIsfSjxRHgFdiAiULreFWIyE  
OeYrMYOcHhiAqFrXmWbUwkOhCTNyCmdrqbJcaxfHFniVTzTFwVATgpaIiOgKajcq1LsljyTOMbJGBDmXzMQFbRn  
kFuVFDXWPJoaSNUGzVqfwUrttekYgblRGFmmjkiYTzZMkTtJOJPlDhEX1PVsQhJxNWJXgRVapUNzOgjCQZNugsr  
simKYut</addit>
```

```
</meet>
```

```
</meetings>
```

Исходные тексты программы существуют в репозитории на github

(<https://github.com/dorogoff/Databases/tree/master/CrmApp>) а так же в svn спбгпу.

```
// Main.java
// главный класс
// Дорогов Алексей, 4081/11
package dorogoff;
/* imports */
public class Main {

    static Connection conn;
    static Driver d;
    static String config[] = new String[6];
    static Statement state;
    static ResultSet resSet;

    public static void main(String argv[]) {
        if (argv.length < 2) {
            System.out.println("Please use -i filename.xml for import; \nPlease use -e
filename.xml for export");
            return;
        }
        // получим найтроски для подключения к базе
        getConfig();

        // создадим подключение
        // jdbcbird скопировать в Java\jdk1.6.0_24\jre\lib\ext
        try {
            Class.forName(config[3]);
            conn = DriverManager.getConnection(config[0], config[1], config[2]);
            if (conn == null) {
                System.out.println("Cant open connection url: " + config[0] + "user: "
+ config[1] + "password: " + config[2]);
                return;
            }
            // установим авто коммит
            conn.setAutoCommit(true);
            state = conn.createStatement();
        } catch (SQLException e) {
            System.out.println("Error: sql : " + e.getMessage());
            return;
        } catch (ClassNotFoundException e) {
            System.out.println("Firebird JCA-JDBC driver not found in class path");
            System.out.println(e.getMessage());
        }
        // Если импорт
        if (argv[0].equals("-i")) {
            importMethod(argv[1].toString());
        }
        if (argv[0].equals("-e")) {
            exportMethod(argv[1].toString());
        }
    }

    // метод для импорта данных из файла указанного в аргументах
```

```

public static void importMethod(String filename) {
    try {
        DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder docBuilder = docBuilderFactory.newDocumentBuilder();
        Document doc = docBuilder.parse(new File(filename));
        doc.getDocumentElement().normalize();
        // будем искать элемент meet
        NodeList listOfMeets = doc.getElementsByTagName("meet");
        int totalRows = listOfMeets.getLength();
        int sum = 0;
        System.out.println("Total rows to import: " + totalRows);
        for (int s = 0; s < totalRows; s++) {
            Node firstPersonNode = listOfMeets.item(s);
            if (firstPersonNode.getNodeType() == Node.ELEMENT_NODE) {
                Element Node = (Element) firstPersonNode;
                String dateText = "", clientId = "", workerId = "", additString =
";
                try {
                    // получим дату
                    NodeList dateList = Node.getElementsByTagName("date");
                    Element dateElement = (Element) dateList.item(0);
                    NodeList textDate = dateElement.getChildNodes();
                    dateText = ((Node) textDate.item(0)).getNodeValue().trim();
                    //получим идентификатор работника
                    NodeList workerList = Node.getElementsByTagName("worker");
                    Element workerElement = (Element) workerList.item(0);
                    NodeList textWorker = workerElement.getChildNodes();
                    workerId = ((Node) textWorker.item(0)).getNodeValue().trim();
                    // идентифкатор клиента
                    NodeList clientList = Node.getElementsByTagName("client");
                    Element clientElement = (Element) clientList.item(0);
                    NodeList textClient = clientElement.getChildNodes();
                    clientId = ((Node) textClient.item(0)).getNodeValue().trim();
                    // дополнительная информация
                    NodeList additList = Node.getElementsByTagName("addit");
                    Element additElement = (Element) additList.item(0);
                    NodeList textAddit = additElement.getChildNodes();
                    additString = ((Node) textAddit.item(0)).getNodeValue().trim();
                } catch (NullPointerException e) {
                    System.out.println("Error: record error, some entries haven't
all info");
                }
                // проверим не назначена ли уже эта встреча
                try {
                    resSet = state.executeQuery("select meet_id from meetings where
cl_man="
                    + clientId + " AND w_man=" + workerId + "");
                    int cnt = 0;
                    while (resSet.next()) {
                        cnt++;
                    }
                    // System.out.println("cnt: " + cnt);
                    if (cnt == 0) {
                        // добавляем запись
                        state.execute("insert into meetings(datum, cl_man, w_man,
addit) "
                        + "values('" + dateText + "', '" + clientId + "',
'" + workerId + "', '" + additString + "')");
                        System.out.println("Record insered.");
                        // инкрементируем общий счетчик добавленных записей
                        sum++;
                    }
                }
            }
        }
    }
}

```

```

        } else {
            System.out.println("Not records to import");
        }
    } catch (SQLException e) {
        System.out.println("Sql error: " + e.getMessage());
    }
}

}

System.out.println("Import finished, added rows: " + sum);
} catch (SAXException e) {
    System.out.println("Error: " + e.getMessage());
} catch (ParserConfigurationException e) {
    System.out.println("Error: " + e.getMessage());
} catch (IOException e) {
    System.out.println("Error: cannot open file" + e.getMessage());
}

}

// экспорт данных о встрече в файл
public static void exportMethod(String filename) {
    // config[4] -начало периода
    // config[5] - конец периода
    // проверим существует ли уже такой файл
    if (new File(filename).exists()) {
        System.out.println("File " + filename + "already exist, you want to
overwrite them?");
        System.out.print("Type yes to overwrite, or type new filename: ");
        // предложим поменять имя файла
        BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in));
        String answer = "";
        try {
            answer = reader.readLine();
        } catch (IOException e) {
            System.out.println("IO error");
        }
        if (!answer.equals("yes")) {
            System.out.println("Your select other file: " + answer);
            filename = answer;
        }
    }
    // формируем запрос на получение встреч
    String queryString = "SELECT * from meetings where datum between '" + config[4]
+ "' AND '" + config[5] + "'";
    try {
        // получим встречи
        resSet = state.executeQuery(queryString);
        OutputStreamWriter out = new OutputStreamWriter(new
FileOutputStream(filename));
        // сформируем xml файл
        out.write("<meetings>\n");
        while (resSet.next()) {
            // если что есть пишем в файл
            out.write("<meet>\n");
            out.write("<t<data>" + resSet.getString(2).substring(0, 19) +
"</data>\n");
            out.write("<t<client>" + resSet.getString(3) + "</client>\n");
            out.write("<t<worker>" + resSet.getString(4) + "</worker>\n");
            out.write("<t<addit>" + resSet.getString(5) + "</addit>\n");
            out.write("</meet>\n");
        }
    }
}

```

```

        out.write("</meetings>\n");
        out.close();
    } catch (SQLException e) {
        System.out.println("SQL Error: ошибка входных данных " + e.getMessage());
    } catch (FileNotFoundException e) {
        System.out.println("Error: file not found");
    } catch (IOException e) {
        System.out.println("Error: " + e.getMessage());
    }
}

// метод для получения настроек к БД из файла, формат:
// url к базе
// username
// password
// драйвер org.firebirdsql.jdbc.FBDriver
// дата начала периода
// дата конца периода
public static void getConfig() {
    try {
        FileInputStream fStream = new FileInputStream("config.txt");
        DataInputStream in = new DataInputStream(fStream);
        BufferedReader buffer = new BufferedReader(new InputStreamReader(in));
        String tmp;
        int cnt = 0;
        while ((tmp = buffer.readLine()) != null) {
            config[cnt] = tmp;
            cnt++;
        }
        in.close();
    } catch (IOException e) {
        System.out.println("Error: cannot open file" + e.getMessage());
    }
}
}

```

Выводы: данная работа позволила изучить работу с firebird из Java с использованием JDBC. В данном приложении нет необходимости в создании и манипулировании объектами, поэтому в данном случае jdbc представляет удобное средство разработки приложений с использованием баз данных. Так же необходимо отметить что JDK представляет удобные средства для парсинга xml данных. В случаях когда имеет смысл пользоваться записями из БД как объектами, или когда много объектов которые необходимо отразить в базу данных можно использовать различные ORM решения (hibernate для Java или circumflex для Scala).