Курс «Базы данных»

Тема: Пример проектирования бизнес-логики.

Барабанщиков
Игорь Витальевич

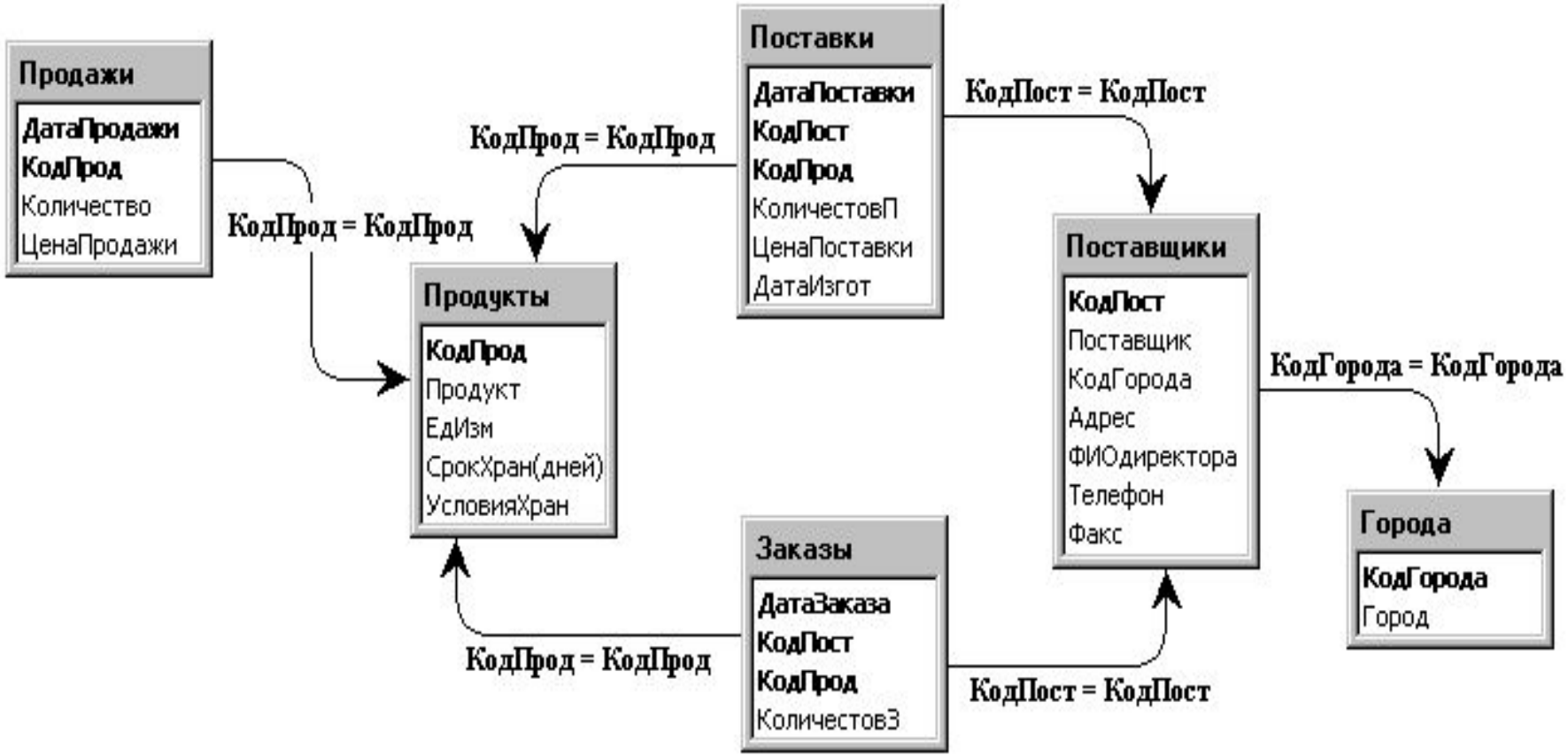
План лекции

- Реализация правил бизнес-логики с помощью триггеров и хранимых процедур для БД «Продажи продуктов».

Денормализация таблицы «Продукты»

Проблема: Для определения возможности продажи любого продукта надо проводить дополнительные вычисления:

- Вычислять общее количество поставленных и проданных продуктов
- Вычислять **ОСТАТОК = ПОСТАВЛЕНО - ПРОДАНО**



Денормализованная таблица

- В процессе физического проектирования таблица «Продукты» была денормализована.
- В таблицу «Продукты» было добавлено поле «Количество».
- Для поддержания согласованности БД надо использовать серверную логику – триггеры или хранимые процедуры.
- Создадим триггеры для таблиц «Поставки» и «Продажи».

Триггер для таблицы «Поставки»

Триггер должен реализовывать следующее бизнес-правило:

- Если количество поставленного продукта –**положительное число, то разрешить вставку записи и обновить остаток** в таблице «Продукты».
- Иначе выдать сообщение об ошибке.

Реализация триггера

```
create or replace trigger fps_tr_supply_ins
   after insert on fps_tt_supply
   for each row
begin
   if :new.quantity  >  0 then
      update fps_ts_product p
         set p.quantity = p.quantity + :new.quantity
         where p.product_id = :new.product_id;
   else
      raise_application_error(-20001,'Недопустимое
            количество');
   end if;
end fps_tr_supply_ins;
```

Триггер для таблицы «Продажи»

Триггер должен реализовывать следующее бизнес-правило:

- Если «Количество» товара в таблице «Товары» больше или равно чем количество продаваемого товара, то разрешить продажу и обновить остаток товара.
- Иначе выдать сообщение об ошибке.

Реализация триггера

```
create or replace trigger fps_tr_sale_ins
  before insert on fps_tt_sale
  for each row
declare
  v_cnt_prod  fps_ts_product.quantity%type;
begin
  select p.quantity into v_cnt_prod
    from fps_ts_product p
    where p.product_id = :new.product_id
    for update nowait;

  if v_cnt_prod >= :new.quantity then
    update fps_ts_product t
      set t.quantity = t.quantity - :new.quantity
      where t.product_id = :new.product_id;
  else
    raise e_invalid_count;
  end if;
end fps_tr_sale_ins;
```

Процедура вставки в таблицу «Поставки»

Процедура должна выполнять проверки:

- Корректности даты поставки – д.б. меньше или равна текущей дате.
- Корректности даты изготовления – д.б. меньше или равна текущей дате.
- Цены продукта – д.б. в диапазоне от 1 до 1000.

Реализация процедуры

```
create or replace procedure supply_ins
 (   p_supply_date  in  date,
    p_provider_id  in  number,
    p_product_id   in  number,
    p_quantity     in  number,
    p_price        in  number,
    p_create_date  in  date)
is
begin
  if  p_supply_date  >  trunc(sysdate)  then
       raise_application_error(-20001,'Неправильная дата поставки');
  end if;
  if  p_supply_date  >  trunc(sysdate)  then
       raise_application_error(-20001,'Неправильная дата изготовления');
  end if;
  if  (p_price < 1)  or  (p_price > 1000)  then
       raise_application_error(-20001,'Неправильная цена');
  end if;
  insert  into  fps_tt_supply(supply_date, provider_id, product_id,
                        quantity, price, create_date)
            values(p_supply_date, p_provider_id, p_product_id,
                  p_quantity, p_price, p_create_date);
  commit;
end  supply_ins;
```

Процедура вставки в таблицу «Продажи»

Процедура должна выполнять проверки:

- Корректности даты продажи – д.б. меньше или равна текущей дате.
- Количества продаваемого продукта – нельзя за одну операцию продавать более 100 единиц товара и менее 1.

Обрабатывать исключительную ситуацию – продажа большего количества продукта, чем имеется в

Реализация процедуры

```
create or replace procedure sale_ins
(  p_sale_date in fps_tt_sale.sale_date%type,
   p_product_id in fps_tt_sale.product_id%type,
   p_quantity   in fps_tt_sale.quantity%type,
   p_price      in fps_tt_sale.price%type)
Is
   e_invalid_count   exception;
begin
   if p_sale_date > trunc(sysdate) then
      raise_application_error(-20001,'Неправильная дата продажи');
   end if;
   if (p_quantity < 1) or (p_quantity > 100) then
      raise_application_error(-20001,'Недопустимое количество');
   end if;
   insert into fps_tt_sale(sale_date, product_id, quantity, price)
      values(p_sale_date, p_product_id, p_quantity, p_price);
   commit;
exception
   when e_invalid_count then
      rollback;
      raise_application_error(-20001,'Недостаточно товара для продажи');
    when others then
       rollback;
end sale_ins;
```

Итоги

- Выполнена реализация бизнес-логики для таблиц «Поставки», «Продукты», «Продажи».
- Аналогично можно реализовать бизнес-логику для других таблиц.
- Например, при уменьшении количества продукта ниже определенного порога – автоматически делать заказ этого продукта.