

# Перезапуск Data Engineer

Решаем тест по предыдущей лекции  
до 18.40

# Организация БД и структуры хранения данных



# Организация БД и структуры хранения данных



**Файл контроля / Controlfile**

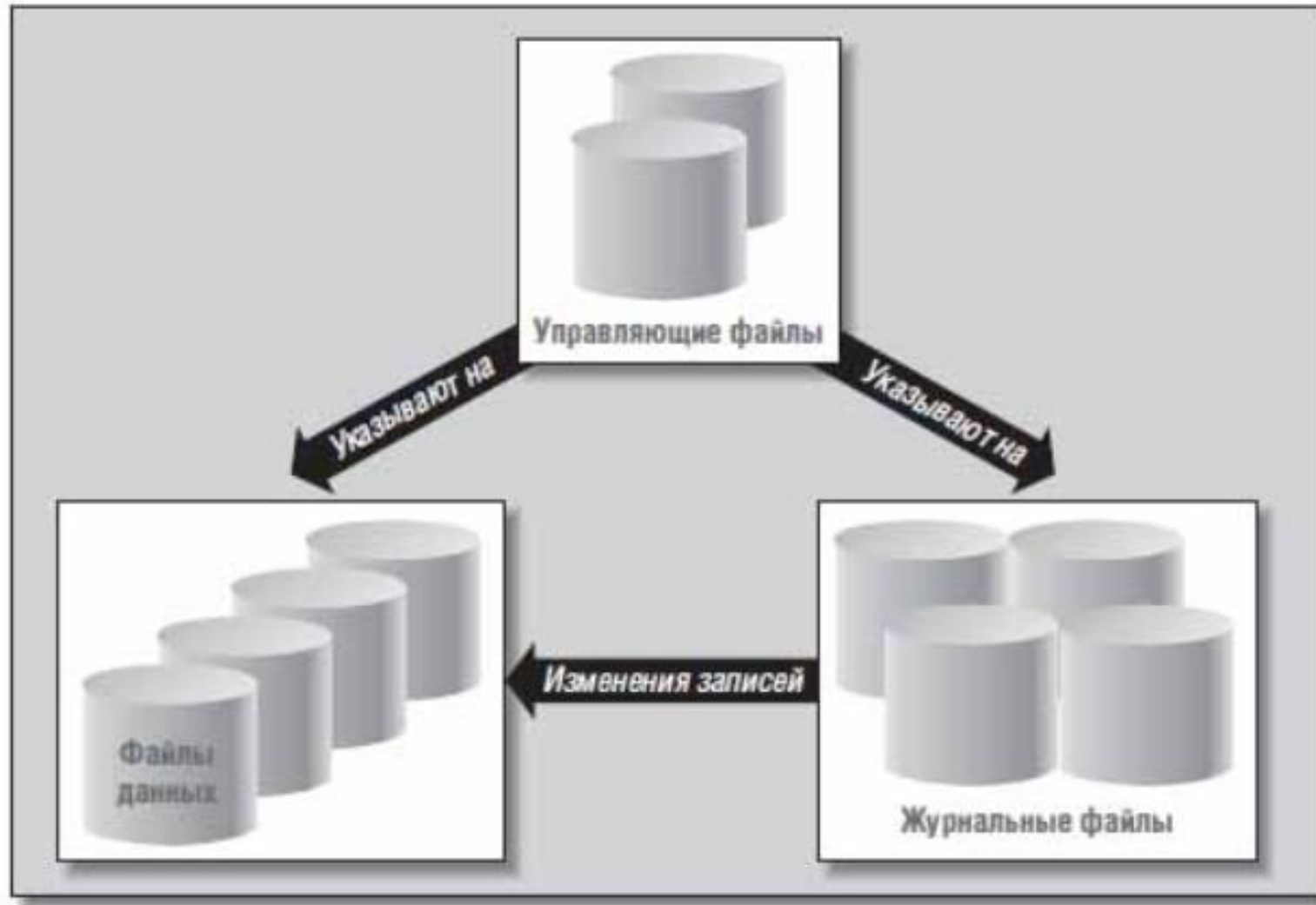


**Файлы логов / Online redo log files**

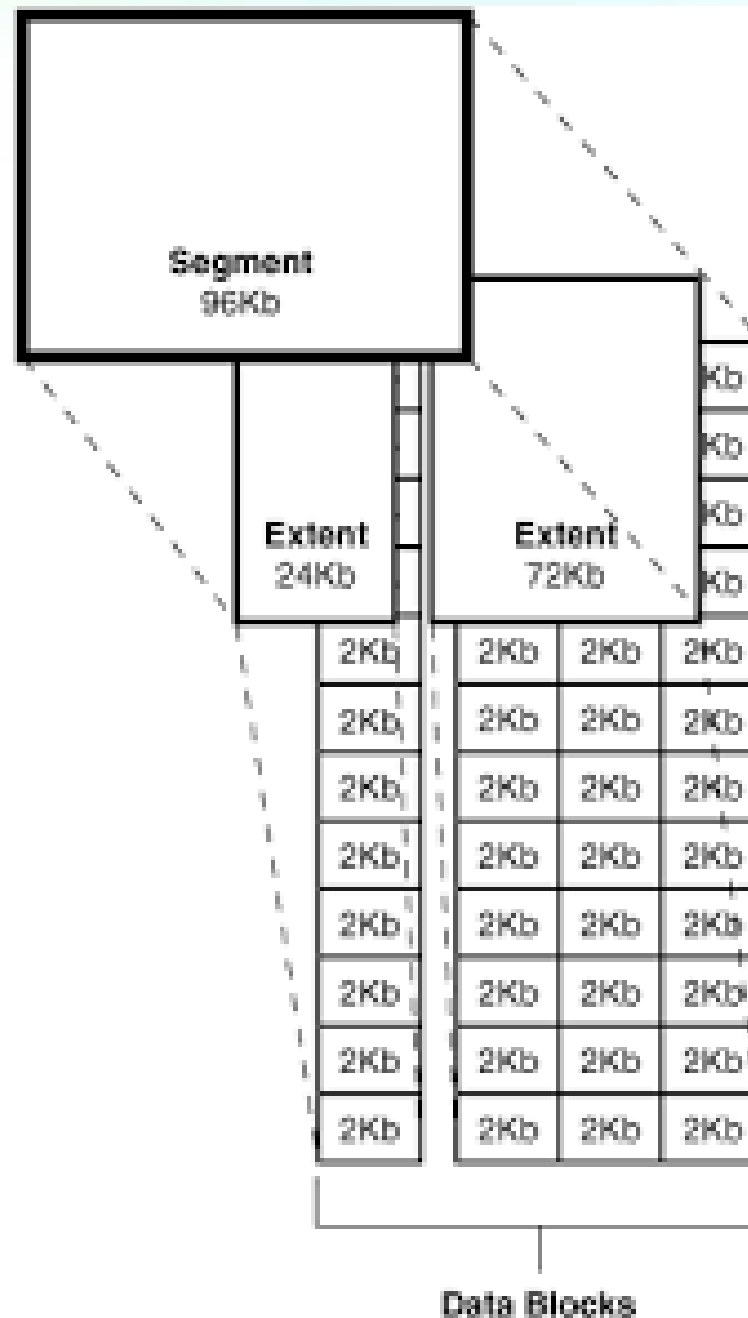


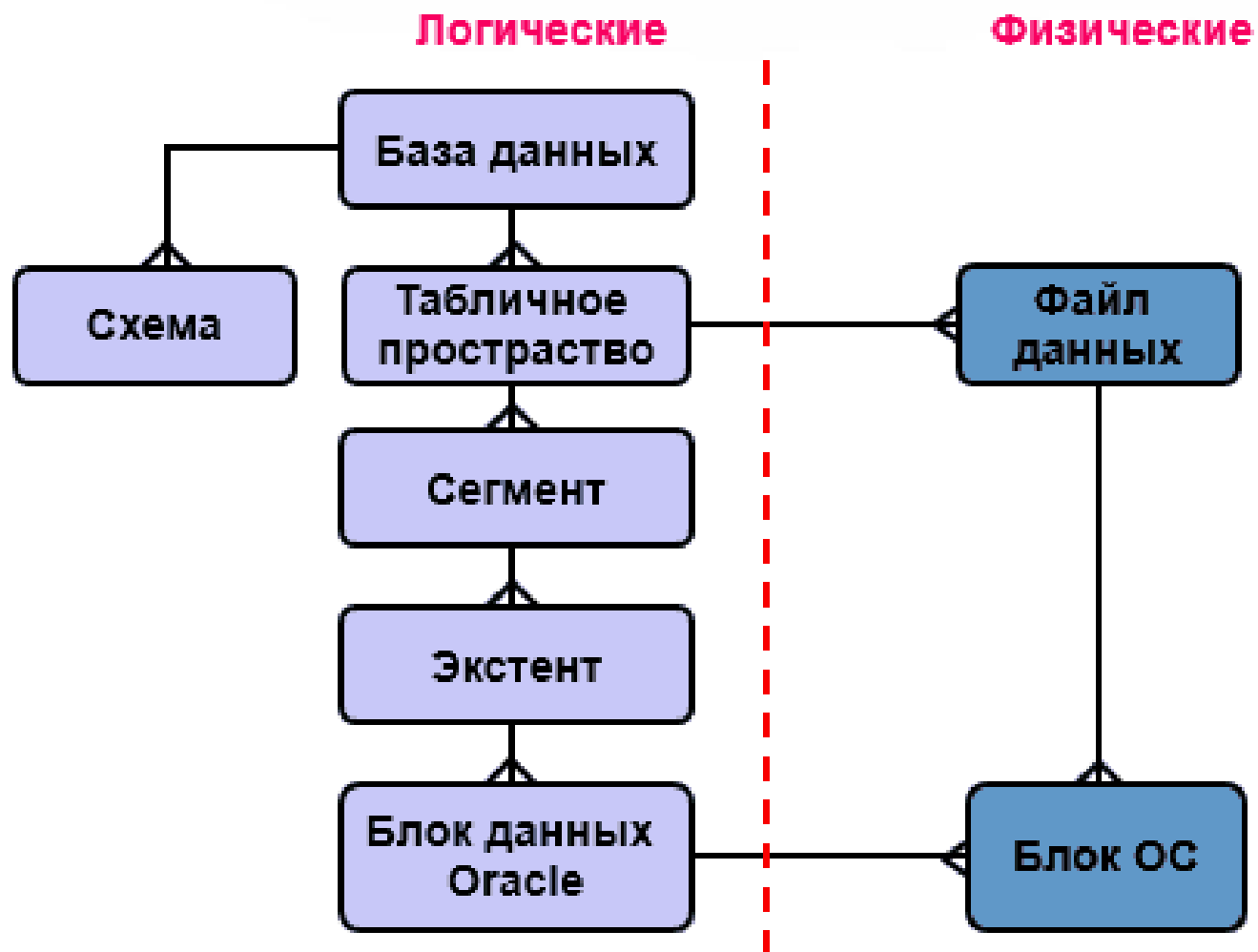
**Файлы данных / Datafiles**

# Организация БД и структуры хранения данных



# Дата файлы





# Словари данных (Метаданные)

Все представления начинающиеся с **USER\_** будут отображать объекты БД принадлежащие текущему пользователю.

Представления начинающиеся с **ALL\_** отображают информацию об объектах к которым у текущего пользователя есть доступ – то есть все объекты которые во создали плюс объекты к которым есть соответствующее разрешение.

Представления с префиксом **DBA\_** отображают информацию о всех объектах БД.



# Объекты в БД

- **Таблицы, представления и синонимы.** Используются для хранения данных и доступа к ним. Основными структурными единицами хранения в Oracle являются таблицы. Представления логически соответствуют подмножествам данных из одной или нескольких таблиц. Синонимы обеспечивают альтернативные имена объектов базы данных.
- **Индексы.** Используются для ускорения доступа к данным.
- **Последовательности.** Используются для генерации чисел для различных целей.
- **Триггеры и ограничения целостности.** Используются для поддержания допустимости вводимых данных.
- **Привилегии, роли и профили.** Используются для управления доступом к базе данных и ее использованием.
- **Пакеты, процедуры и функции.** Прикладные коды PL/SQL, используемые в базах данных.

# GRANT – Права

- GRANT privileges ON object TO user;
- REVOKE privileges ON object FROM user;

Привилегии	Описание
SELECT	Возможность выполнения SELECT на таблице
INSERT	Возможность выполнения INSERT на таблице
UPDATE	Возможность выполнения UPDATE на таблице
DELETE	Возможность выполнения DELETE на таблице
REFERENCES	Возможность создавать CONSTRAINT, который ссылается на таблицу.
ALTER	Возможность выполнять оператор ALTER TABLE, чтобы изменить описание таблицы.
INDEX	Возможность создавать INDEX таблице с помощью оператора CREATE INDEX.
ALL	Все привилегии для таблицы

# Роли

```
CREATE ROLE role_name  
[ NOT IDENTIFIED |  
IDENTIFIED {BY password | USING [schema.] package | EXTERNALLY |  
GLOBALLY }];
```

Example: CREATE ROLE test\_role IDENTIFIED BY password;

```
GRANT privileges ON object TO role_name;  
GRANT EXECUTE ON object TO role_name;
```

```
GRANT select, insert, update, delete ON suppliers TO test_role;
```

```
GRANT role_name TO user_name;
```

```
GRANT test_role TO vasya;
```

# CREATE TABLE

```
CREATE TABLE COUNTRIES
```

```
(  
  COUNTRY_ID CHAR (2 BYTE) NOT NULL ,  
  COUNTRY_NAME VARCHAR2 (40) ,  
  REGION_ID NUMBER  
)
```

```
;
```

```
ALTER TABLE COUNTRIES
```

```
  ADD CONSTRAINT COUNTRY_C_ID_PK PRIMARY KEY (  
  COUNTRY_ID );
```

```
ALTER TABLE COUNTRIES
```

```
  ADD CONSTRAINT COUNTR_REG_FK FOREIGN KEY  
(REGION_ID) REFERENCES REGIONS( REGION_ID) NOT  
DEFERRABLE
```

```
;
```

```
CREATE TABLE COUNTRIES
```

```
( COUNTRY_ID CHAR (2 BYTE) NOT NULL  
  , COUNTRY_NAME VARCHAR2 (40 BYTE)  
  , REGION_ID NUMBER  
  , CONSTRAINT COUNTRY_C_ID_PK PRIMARY KEY  
(COUNTRY_ID)  
  , CONSTRAINT COUNTR_REG_FK FOREIGN KEY  
(REGION_ID) REFERENCES REGIONS(REGION_ID)  
);
```

# Какие еще constraint бывают

```
ALTER TABLE EMPLOYEES ADD CONSTRAINT EMP_EMAIL_UK UNIQUE ( EMAIL );
```

```
ALTER TABLE employees ADD CONSTRAINT DF_Com DEFAULT 0,1 FOR comission_pct;
```

```
CREATE TABLE suppliers  
(  
    supplier_id numeric(4),  
    supplier_name varchar2(50),  
    CONSTRAINT check_supplier_id  
    CHECK (supplier_id BETWEEN 100 and 9999)  
);
```

```
ALTER TABLE suppliers ADD CONSTRAINT check_supplier_name CHECK (supplier_name IN ('Apple',  
'DELL', 'Nec'));
```

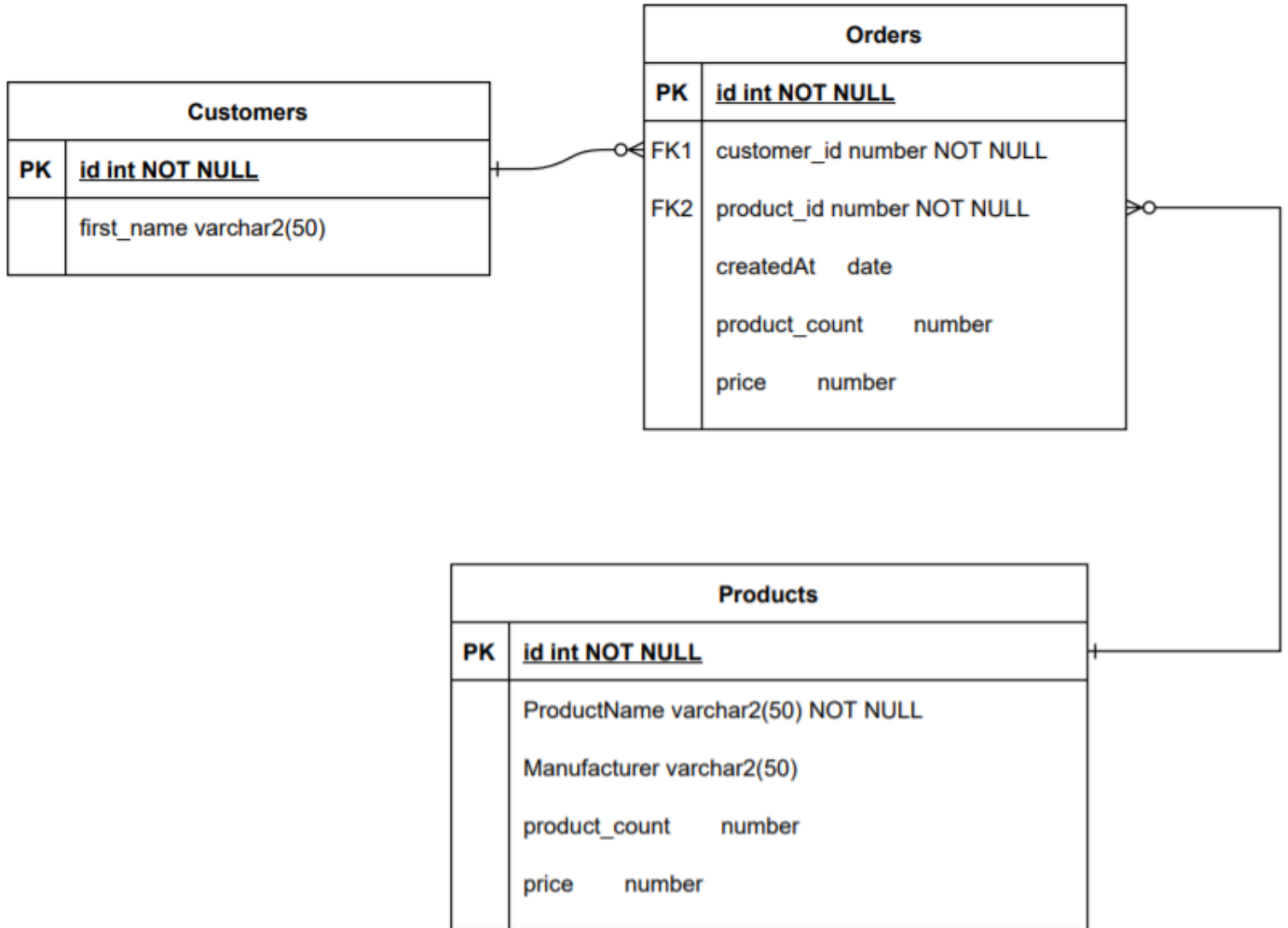
```
ALTER TABLE table_name ADD CONSTRAINT constraint_name CHECK (column_name condition)  
[DISABLE];
```

Ключевое слово DISABLE является необязательным. Если вы создаете проверку ограничений с помощью ключевого слова DISABLE, ограничение будет создано, но условие не будет принудительно выполнено.

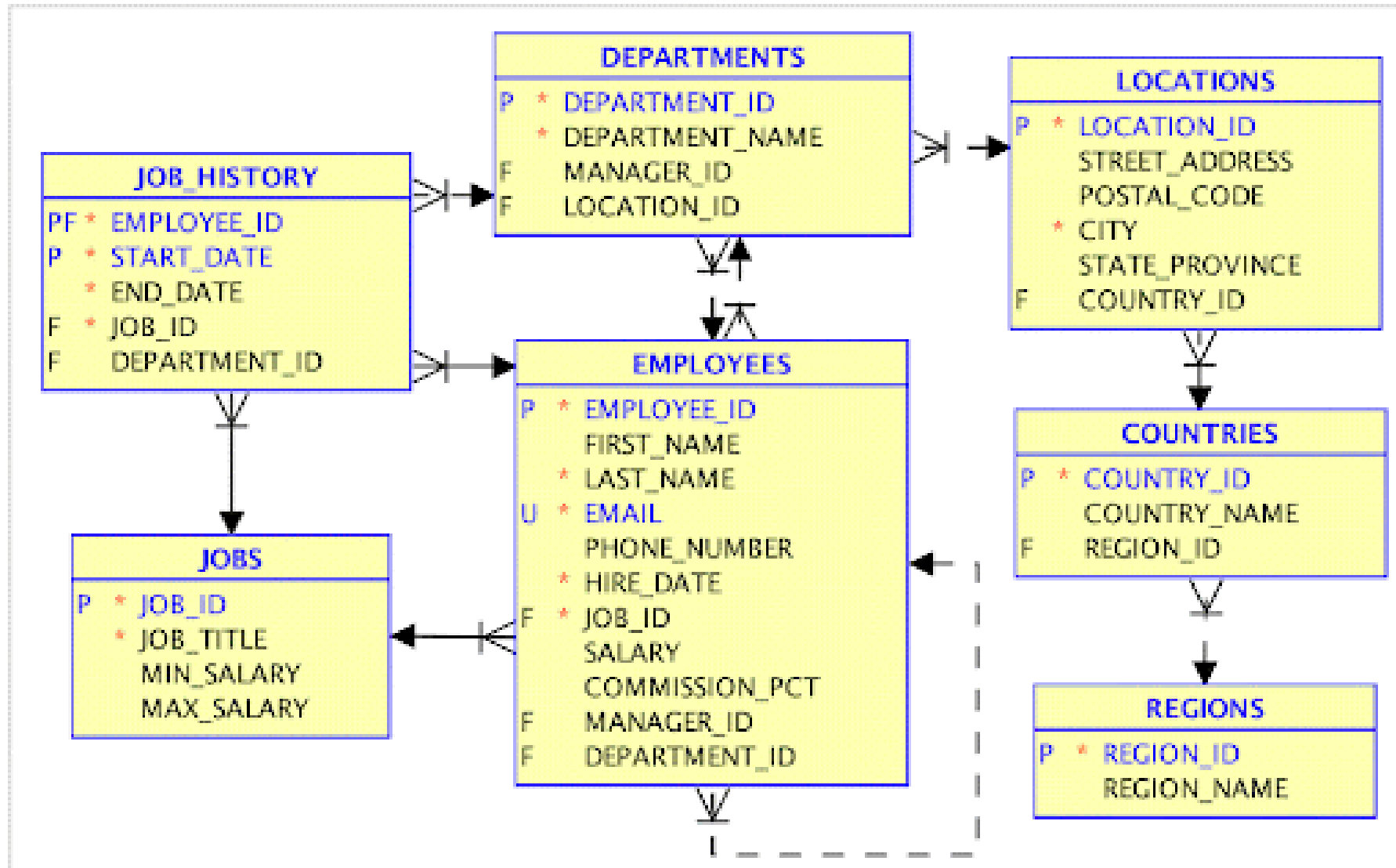
# Primary and foreign keys

После создания первичного ключа  
БД сразу же создает индекс и ставит  
ограничение not null

Значения внешнего ключа являются  
подмножеством первичного ключа, к  
которому относятся



# Primary and foreign keys



# Нормальная форма

свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. **Нормальная форма** определяется как совокупность требований, которым должно удовлетворять отношение



# КЛАССЫ ОТНОШЕНИЙ



Взаимосвязь нормальных форм

# Ненормализованная БД

Ключ	Сотрудник	Должность	Отдел	Телефоны
1	Иван	Программист	IT	
2	Марья	Секретарь	Инфраструктура	123,456

# 1НФ

- Порядок строк и столбцов неважен
- Нет повторяющихся строк
- Каждая ячейка таблицы имеет только одно значение из соответствующего домена

Ключ	Сотрудник	Должность	Отдел	Телефоны
1	Иван	Программист	IT	
2	Марья	Секретарь	Инфраструктура	123
3	Марья	Секретарь	Инфраструктура	456

# 2НФ

- Исполняются все условия для 1НФ
- Каждый неключевой атрибут зависит от ключа

Ключ	Сотрудник	Должность	Отдел
1	Иван	Программист	IT
2	Марья	Секретарь	Инфраструктура
Ключ сотрудника	Телефон		
2	123		
2	456		

# 3НФ

- Исполняются все условия для 2НФ
- Каждый неключевой атрибут должен предоставлять информацию о ключе, полном ключе и ни о чем, кроме ключа

Ключ	Сотрудник	Ключ должности
1	Иван	1
2	Марья	3
Ключ сотрудника	Телефон	
2	123	
2	456	
Ключ должности	Должность	Отдел
1	Программист	IT
2	Системный администратор	IT
3	Секретарь	Инфраструктура
4	Бухгалтер	Инфраструктура

# Партицирование таблиц

- By list – по конкретному перечню дискретных значений столбца
- By hash – равномерное распределение строк по партициям по алгоритмам хеширования Oracle
- By range - по диапазону дат и по диапазону значений

# Партицирование таблиц

## ***Range-partitioned table***

```
CREATE TABLE sales (  
  prod_id NUMBER(6)  
  , cust_id NUMBER  
  , time_id DATE  
  , channel_id CHAR(1)  
  , promo_id NUMBER(6)  
  , quantity_sold NUMBER(3)  
  , amount_sold NUMBER(10,2)  
)  
PARTITION BY RANGE (time_id) (  
  PARTITION sales_q1_2006 VALUES LESS THAN (TO_DATE('01-APR-2006','dd-MON-yyyy')) TABLESPACE tsa  
  , PARTITION sales_q2_2006 VALUES LESS THAN (TO_DATE('01-JUL-2006','dd-MON-yyyy')) TABLESPACE tsb  
  , PARTITION sales_q3_2006 VALUES LESS THAN (TO_DATE('01-OCT-2006','dd-MON-yyyy')) TABLESPACE tsc  
  , PARTITION sales_q4_2006 VALUES LESS THAN (TO_DATE('01-JAN-2007','dd-MON-yyyy')) TABLESPACE tsd  
);
```

# Партицирование таблиц

## *hash-partitioned table*

```
CREATE TABLE dept (  
  deptno NUMBER  
  , deptname VARCHAR(32))  
PARTITION BY HASH(deptno)  
PARTITIONS 16;
```



# Партицирование таблиц

## *list-partitioned table*

```
CREATE TABLE q1_sales_by_region (  
  deptno number  
  , deptname varchar2(20)  
  , quarterly_sales number(10, 2)  
  , state varchar2(2)  
  )  
PARTITION BY LIST (state) (  
  PARTITION q1_northwest VALUES ('OR', 'WA')  
  , PARTITION q1_southwest VALUES ('AZ', 'UT', 'NM')  
  , PARTITION q1_northeast VALUES ('NY', 'VM', 'NJ')  
  , PARTITION q1_southeast VALUES ('FL', 'GA')  
  , PARTITION q1_northcentral VALUES ('SD', 'WI')  
  , PARTITION q1_southcentral VALUES ('OK', 'TX')  
  );
```