

# DB期末选择背诵材料

## 知识大全

### 1. 英文缩写

DB 数据库, DBMS(engine)数据库管理系统, DBAP数据库应用程序, DBA数据库管理员, DB user数据库用户

### 2. DBMS数据库的作用：

High data share 高数据共享

Low data **redundancy** 数据冗余度低

allow data independent from program, 允许数据独立于程序

Integrity control, 完整性控制

security control, 安全控制

allow transaction **concurrency**, 允许事务并发

provide backup and recovery, 提供备份和回复

automatic query optimization 自动查询优化

### 3. Data model comprises:

a. A structural part

b. Possibly a set of integrity rules

c. A **manipulative** part

### 4. 关系模型

a. 使用一系列表格来表示数据及其之间的关系。 (比网络或层次模型更简单) 在逻辑和视图层面  
描述数据，抽象了数据存储的低级细节。

b. 定义数据完整性约束规则

c. 定义用于查询数据的数据操作语言

### 5. DDL (关系定义语言)

a. 定义模式/外模式/内模式

- Create/ Drop/Alter table;
- Create/ Drop/Alter **view**;
- Create/ Drop index;

## b. 定义完整性约束：

- i. 数据类型
- ii. 主键外键
- iii. Not null, unique
- iv. 检查；约束；断言；
- v. 事务(ACID)

## c. 定义安全性约束



1. grant DELETE on SC to dean.yan
2. revoke DELETE on SC from dean.yan

## 6. DML数据操作语言

### a. 修改

#### 代码块

```
1  INSERT INTO Employee (ID, name, position) VALUES (104, 'Michael',
  'engineer');
2  UPDATE Employee SET position = 'manager' WHERE ID = 101;
3  DELETE FROM Employee WHERE ID = 103;
```

- i. Delete from <table> ...
- ii. Insert into <table> values (...)
- iii. Update <table> set <attribute>=...

### b. 查询 (SELECT)

- i. 基本查询结构
- ii. Set 操作
- iii. 聚合操作
- iv. 嵌套子查询

## 7. View 视图

### a. 视图 (View) 就是一个虚拟的表，它本身不存储任何数据。它就像一个“窗户”，通过这个窗户，你可以看到你感兴趣的数据，而这些数据实际上来自于一张或多张真实的物理数据表。

- b. 视图的内容是由一个 SQL 查询语句（`SELECT` 语句）定义的。当你查询一个视图时，数据库会实时执行这个预设好的 `SELECT` 语句，并把结果动态地呈现给你，看起来就像你在查询一张真实的表。**`create view v as <查询表达式>`**
- c. permits users to access data in a customized way. 视图允许用户以自定义的方式访问数据。
- d. can simplify complex operations. 简化复杂操作
- e. can hide parts of database from certain users. 向某些用户隐藏数据库的部分内容

## 8. 各个子句的执行顺序：

- **`Order by` clause execution after “select” clause?** No, 执行确在select 之前

代码块

```
1  from clause  
2  where clause  
3  group by clause  
4  having clause  
5  select clause
```

## 9. 数据库应用系统开发阶段

- a. 需求分析
  - i. 需求描述（数据流图+数据字典）
  - ii. **数据需求建模：概念模型ER图设计**
  - iii. 功能需求建模：用例图
- b. 设计
  - i. **数据逻辑结构建模：如关系模式设计以及规范化**
  - ii. 数据物理结构建模：关系模型内模式的设计
  - iii. 功能设计建模：架构、类图、活动图
- c. 实现
  - i. **DDL创建模式（数据结构和各种约束定义）**
  - ii. **DML代码装入初始化数据**
  - iii. **数据库应用程序编码测试**：增删改查等访问操作，数据修改后的完整性检验

## 10. 概念模型是高层次数据抽象的描述，不依赖于具体数据库类型（关系或非关系数据库模型）

- 它抽象，完整的表达数据存储需求（信息完整）
- 它没有冗余表达，一个概念在图中只能出现一次（即：一个属性只出现在一个地方）

## 11. 规范化过程

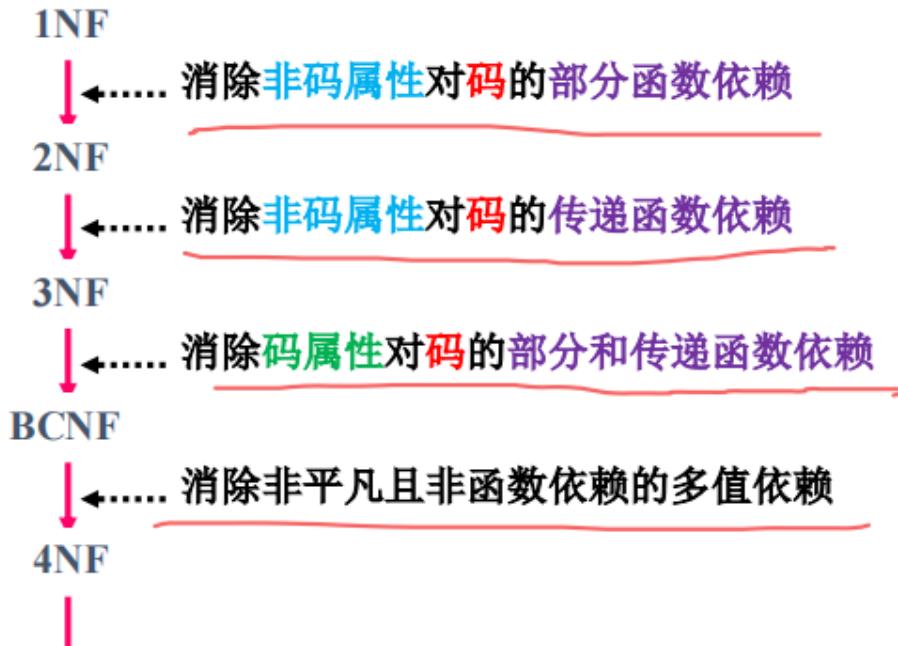
## a. 范式

	违反规则	定义	违反的例子
2NF	存在某个 <b>非主属性</b> ，它只依赖于这个组合候选码的 <b>一部分</b>	所有 <b>非主属性都完全函数依赖于 R</b> 的每一个候选码	候选码AB，非主属性CD AB->C B->D
3NF	存在 <b>非主属性传递依赖于候选码时</b>  存在非主属性依赖于不是候选键时  既左侧不是候选键，又右侧不是主属性	R 中没有 <b>非主属性传递函数依赖于 R 的任何候选码</b>  满足两个条件之一： a) $X$ 是一个 <b>超码 (Superkey)</b> 。 b) $Y$ 中 <b>所有属性都是主属性</b> 。	A是候选码， A->B A->C C->D，导致A->C->D， D 传递依赖于A
BCNF	函数依赖 $X \rightarrow Y$ ， $X$ 不是超码	每个 <b>非平凡的函数依赖 <math>X \rightarrow Y</math></b> ， $X$ 都必须是 R 的一个 <b>超码</b>	AB是候选码。 AB -> CD， D->A， D不是超码
4NF	存在 $X \rightarrow\rightarrow Y$ ， $X$ 不是超码时	每个 <b>非平凡的多值依赖 <math>X \rightarrow\rightarrow Y</math></b> ， $X$ 都必须是 R 的一个 <b>超码</b> 。	课程 $\rightarrow\rightarrow$ 授课老师， 课程 $\rightarrow\rightarrow$ 参考书，课程不是超码

i. 1NF：确保表中的每个单元格都是**原子值**，不可再分。*Excel单元格里不能有“,”隔开的多个值*。

- (注：码属性是包含在“候选码”中的属性，也称为“主属性”)

消除“决定因素非码”的非平凡函数依赖



- ii. 2NF: 表里的所有“非码属性”（普通信息）必须完全依赖于整个“主键”，而不能只依赖主键的一部分。【复合主键时才有意义】——当右边不是主属性的时候，左边必须是主键
  - iii. 3NF: 消除非码属性对码的传递函数依赖。——二满足其一①  $A \rightarrow B$  ( $B$ 不是 $A$ 的一部分)， $A$ 本身必须是一个候选码。②右边是主属性
  - iv. BCNF: 消除主属性对码的部分和传递依赖。—— $A \rightarrow B$  中，只要 $B$ 不是 $A$ 的一部分，那么 $A$ 本身必须是一个候选码。
  - v. 4NF: 消除非平凡且非函数依赖的多值依赖。——一个表里不能有两个或以上相互独立的“一对多”关系。
- b. 函数依赖
  - c. 无损分解

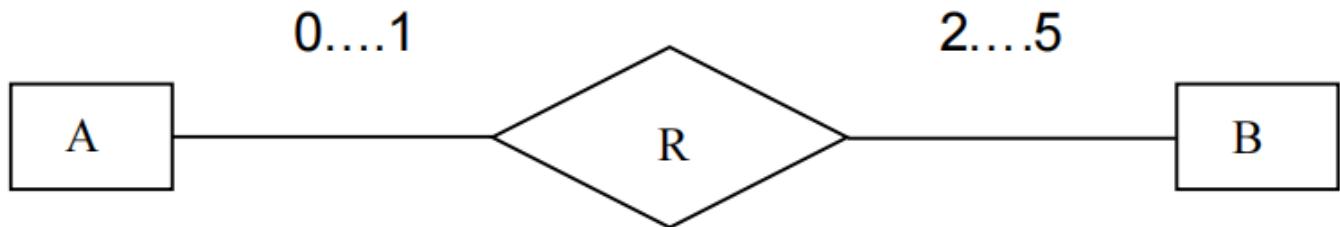
## 选择原题

In levels of Consistency of Transaction in SQL-92, “only committed records to be read, and repeated reads of same record must return same value” is ( REPEATABLE READ )

2020~2021A

1. The degree of a table is the number of ( columns ) in the table.
2. The following are functions of a DBMS are ( 数据库的创建、管理、处理数据、权限管理 ), except (creating and processing forms)
3. (distinct) can be used to take out repetition tuples.

4. To authorize query privilege of tables to database user Wang, and permit Wang to authorize the privilege to others, the SQL sentence is (GRANT SELECT ON S TO Wang WITH GRANT OPTION )
5. SQL **views** can be used to hide: (列和行, 复杂查询语句 )
6. one-to-many



7. 2PL protocol, at (Growing phase) stage, A transaction may obtain locks, but cannot release locks. 【Shrinking phase则相反】
8. In a database system, (the DBMS ) provides data consistency



DBMS 提供的机制包括：

1. 事务管理（Transaction Management）：ACID
2. 并发控制（Concurrency Control）如两阶段锁（2PL）、时间戳排序等机制，防止脏读、不可重复读等现象。
3. 恢复机制（Recovery Mechanism）：基于日志
4. 完整性约束（Integrity Constraints）：主键、外键、CHECK 约束等防止非法数据进入，维护结构上的一致性。

DBA 定义规则，负责的是数据库的配置、权限管理、安全策略、性能优化等。

## 2020~2021B

1. The most widely used data model in the database domain is ( relational model )
2. The foreign key is defined mainly to guarantee the ( Reference Integrity ) of data
3. Creation of an ER diagram is a result in the ( conceptual-design ) phase of database design
4. If all candidate keys for a relation schema consist of only one attribute, the schema is satisfied at least (2NF)
5. A lossless join decomposition of a relation means: the natural join of the relations in the decomposition produces the original relation
6. Dependency Preservation means: no functional dependencies are lost
7. (REVOKE) can be used to remove access privileges associated with a table

## 8. Which pair of locks is compatible:( IX, IX )

A. S, X: 不兼容。读和写是互斥的。

B. X, X: 不兼容。两个写操作不能同时进行。

C. S, IX: 不兼容。如果一个事务准备要写 (IX) , 就不允许另一个事务对整个表加读锁 (S) 。

D. IX, IX: 兼容。两个不同的事务可以同时打算对同一个表中的不同行进行写操作。意向锁只在表级别表示一个意图，真正的冲突会在行级别的X锁上发生。因此，两个IX锁在表级别上是兼容的。

已持有的锁 \ 请求的锁	IS (意向共 享)	IX (意向排 他)	S (共享)	SIX (共享意 向排他)	X (排他)
IS (意向共 享)	兼容	兼容	兼容	兼容	不兼容
IX (意向排 他)		兼容	不兼容	不兼容	不兼容
S (共享)			兼容	不兼容	不兼容
SIX (共享意 向排他)				不兼容	不兼容
X (排他)					不兼容

## 2019-2020B

1. DECLARE CURSOR 声明游标

2. UNIQUE 索引的作用是确保被索引的列中的所有值都是唯一的

3. 查找数据库中所有的数据表用SHOW TABLES

## 2018~2019B

### 2017~2018AB

1. A relation schema R is in 1NF if the domains of all attributes of R are atomic. 第一范式 (1NF) 的定义，即属性值不可再分。
2. Decomposition of a 3NF relation into BCNF always **NOT** preserves functional dependencies. 将一个 3NF 关系分解为 BCNF 总不能保持函数依赖。——虽然任何关系都可以被无损连接地分解为 BCNF，但这个分解过程**不一定能**保持所有的函数依赖。

