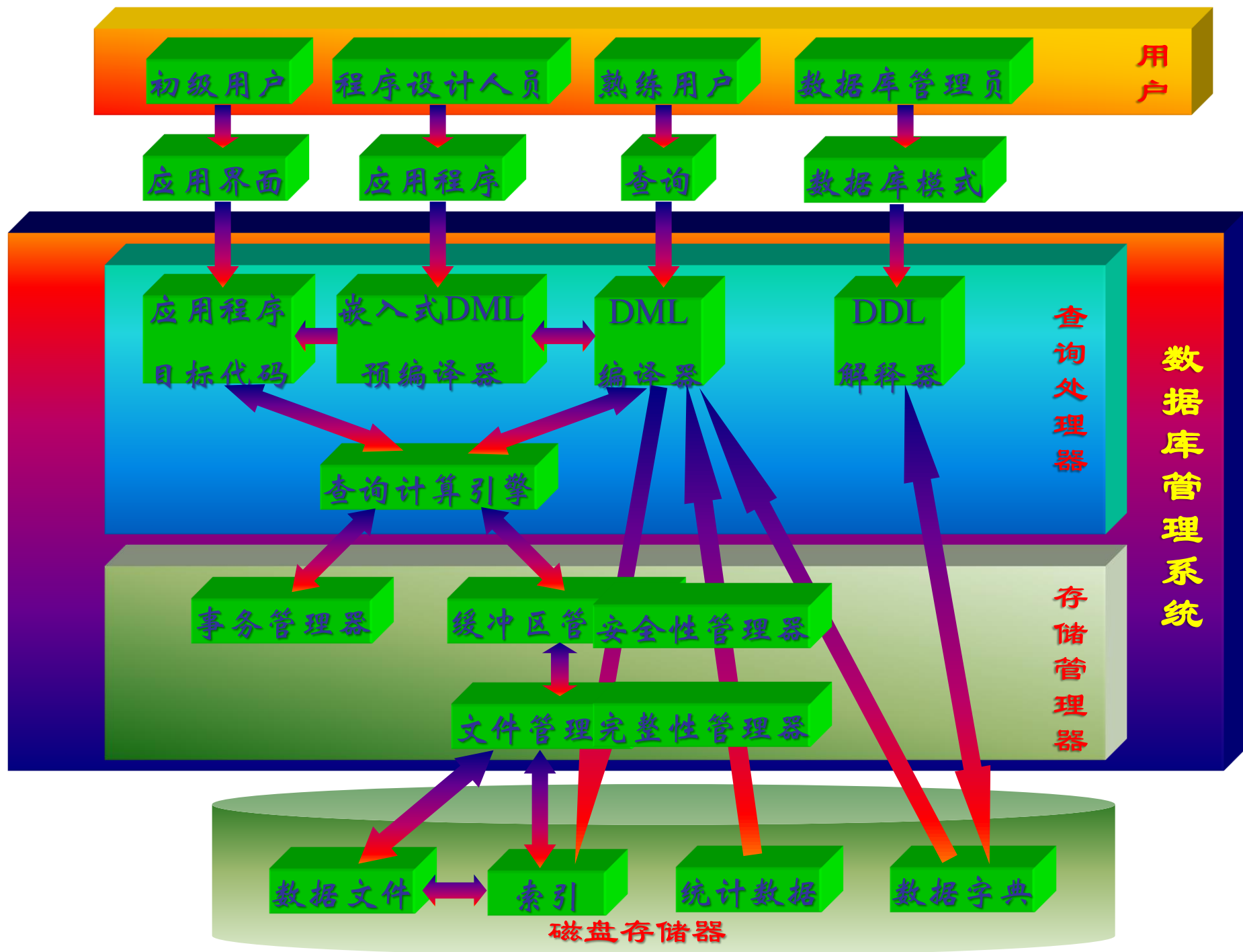
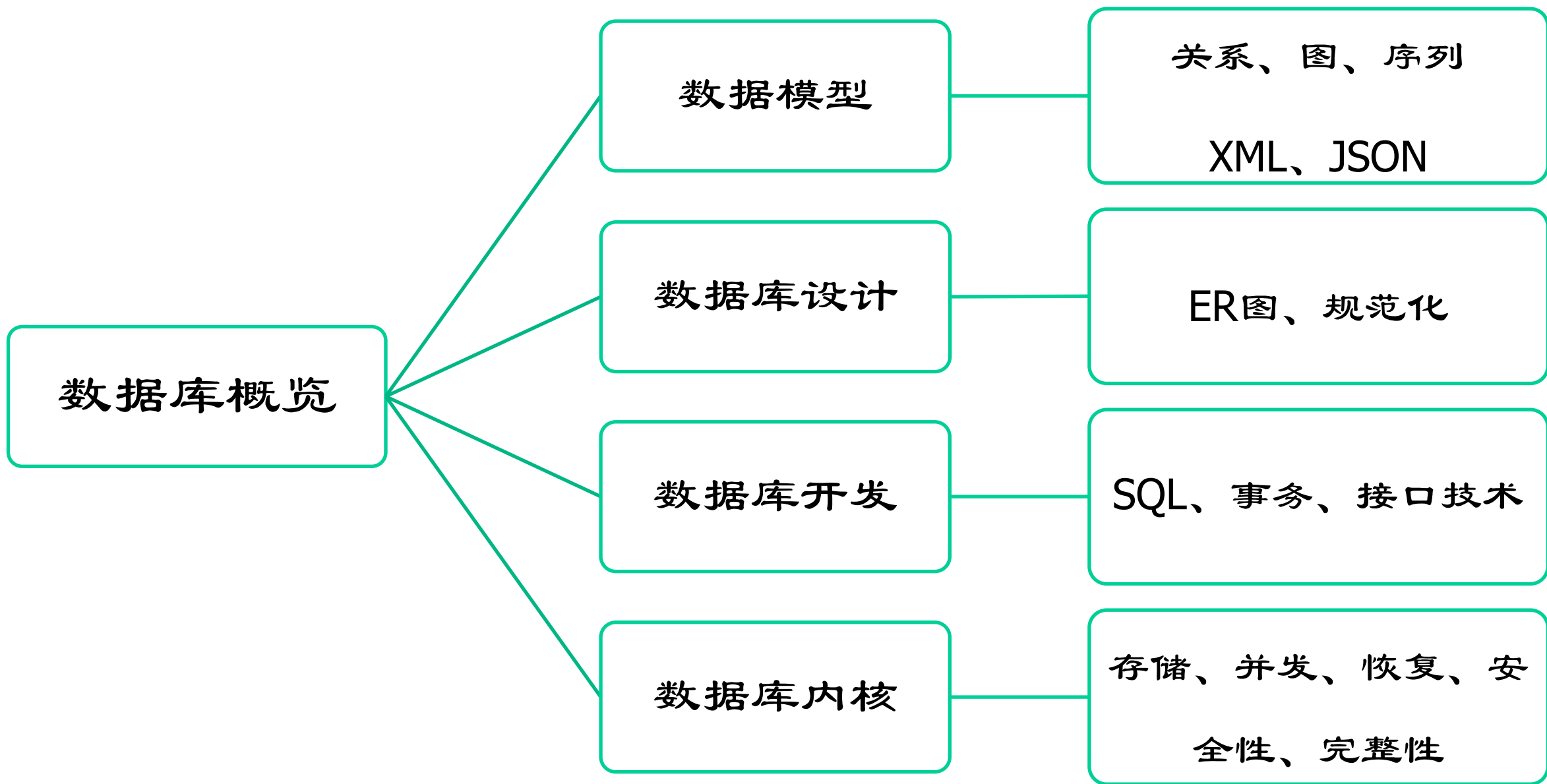


数据库基础知识

数据库系统结构



一般数据库课程覆盖的知识点



关系模型的定义

老师集合 $T = \{t_1, t_2\}$

学生集合 $S = \{s_1, s_2, s_3\}$

课程集合 $C = \{c_1, c_2\}$

笛卡尔积 $T \times S \times C$:
可能的世界

老师	学生	课程
t_1	s_1	c_1
t_1	s_1	c_2
t_1	s_2	c_1
...
t_2	s_3	c_2

Teach(老师、学生、课程)
Assistant(老师、学生、课程)

属性

关系：笛卡尔积
中有意义的子集

元组

老师	学生	课程
t_1	s_1	c_1
t_1	s_1	c_2
t_1	s_2	c_1
t_2	s_3	c_2

关系模型的操作

r

A	B
α	1
β	2

s

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

$r \times s$

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

$\sigma_{A=C}(r \times s)$

A	B	C	D	E
α	1	α	10	a
β	2	β	20	a
β	2	β	20	b

$\Pi_{B,D}(\sigma_{A=C}(r \times s))$

B	D
1	10
2	20

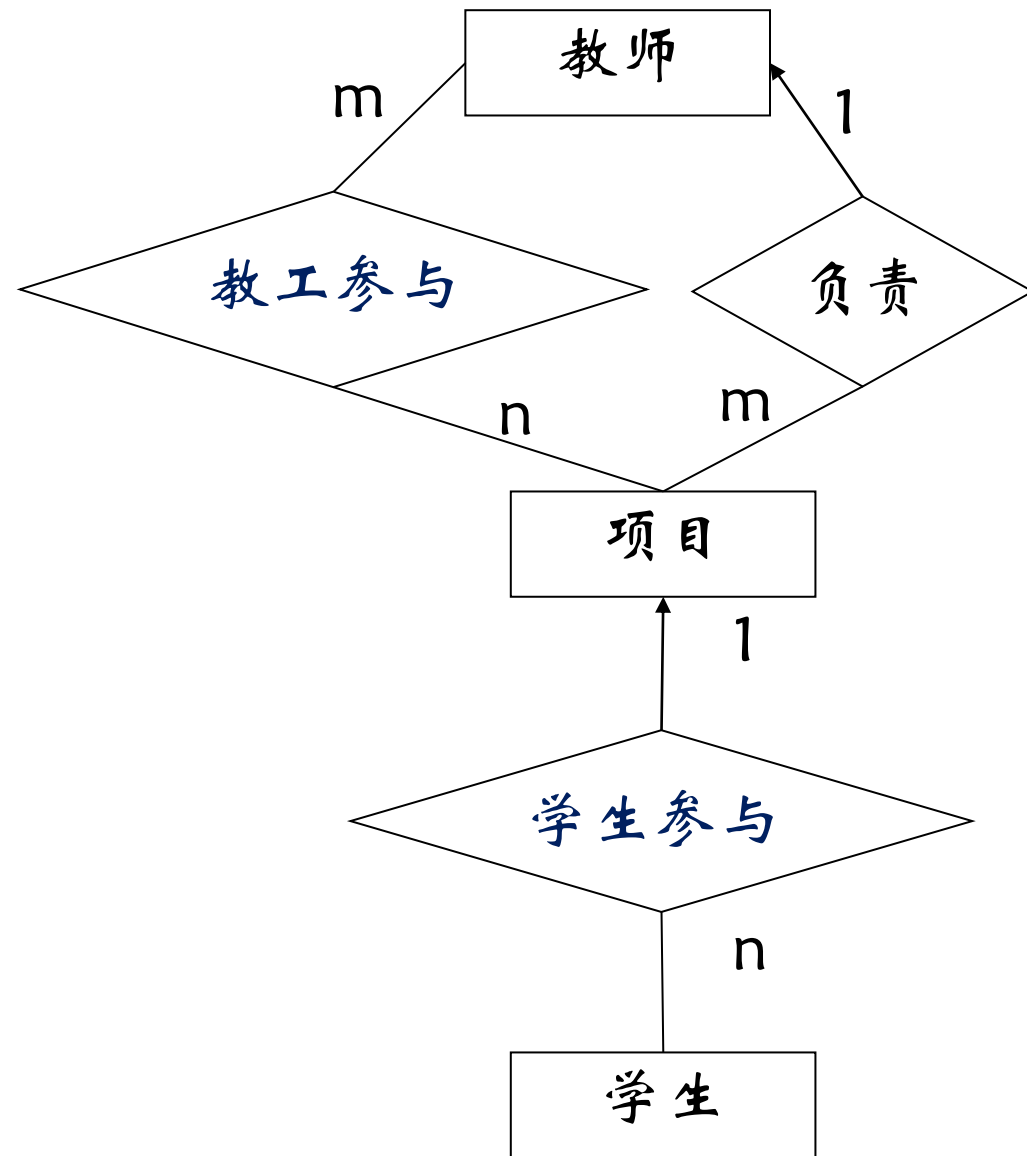
包打天下：

选择、投影、笛卡尔积

基于业务描述的ER设计实例

考虑一个学校数据库，它要存储以下信息：

- 教师： 教工号、教工名、职称；
- 项目： 项目号、项目名称、起始年份、资助额；
- 学生： 学号、学生名、年龄、学位。
- 一个教工可以负责多个项目；
- 每个项目只能有一个负责人；
- 一个老师可以参与多个项目；
- 一个学生只能参与一个项目；
- 一个项目可以有多个学生和老师参与。



由业务单据生成ER模型

卖家信息

昵称：米开朗旗舰店

真实姓名：义乌市美智朗工艺品有限公司

城市：浙江 金华

联系电话：

邮件：m***

支付宝：4***

订单信息

订单编号： 1149473762988781 

支付宝交易号：
2015071021001001130292410113

成交时间： 2015-07-10 16:25:05

付款时间： 2015-07-10 16:25:56

确认时间： 2015-07-21 08:51:32

宝贝	宝贝属性	状态	单价(元)	数量	优惠	商品总价(元)	运费(元)
 米开朗 纯净的理想 木质成人拼图1000片拼图儿童益智玩具情人礼物    	-	已确认收货	128.00	1	-	215.00 2860天猫积分 抵28.60元	0.00 (快递)
 米开朗木质成人拼图500/1000/1500/2000片静静的祝福儿童益智玩具    	颜色分类: 晚安 静静地祝福木质 500片	已确认收货	176.00	1	-		

(可获返还商城积分151点) 

实付款： **186.40** 元

物流信息

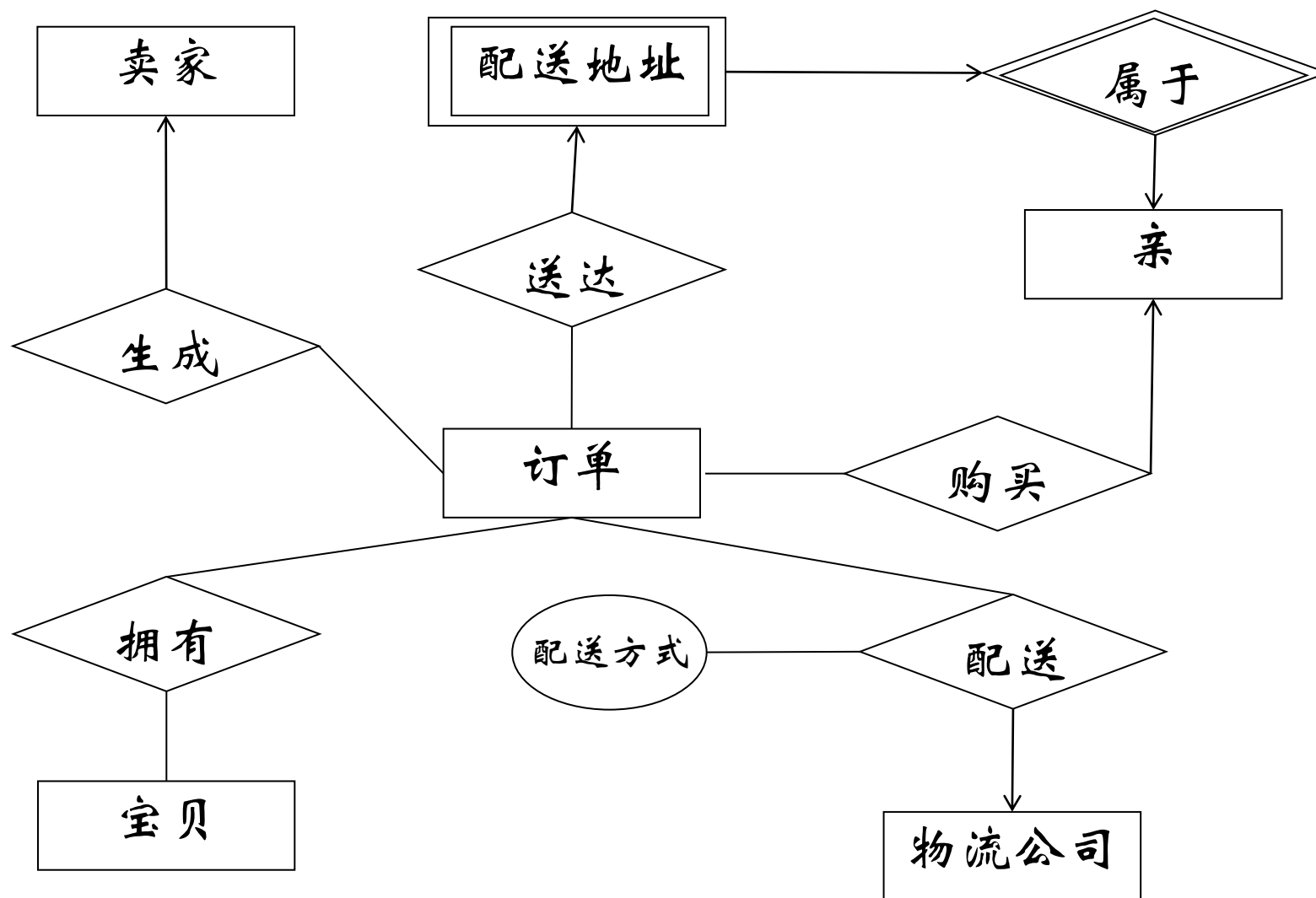
收货地址： 陈立军， 13051836828， ，北京 北京市 海淀区 北京大学理科一号楼1628， 100871

运送方式： 快递

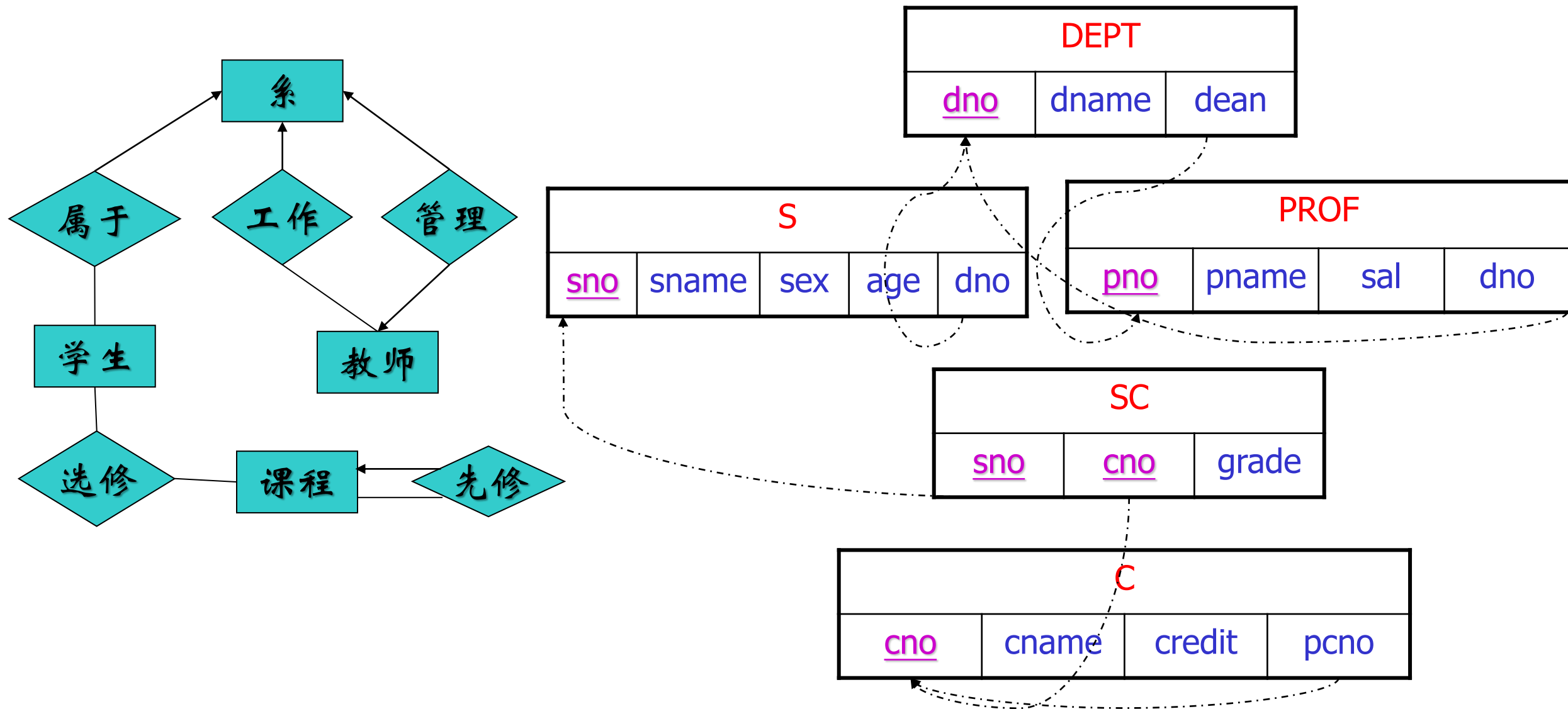
物流公司： 圆通速递



由业务单据生成ER模型



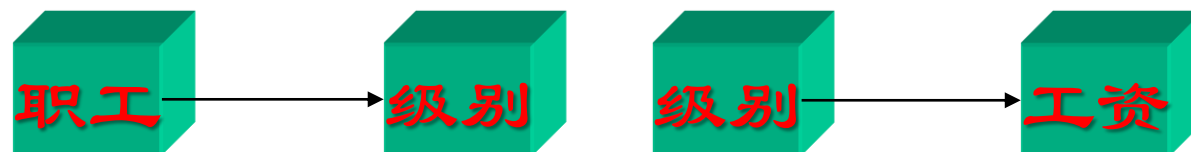
ER图转换为关系模型



关系规范化：消除关系中的不良数据依赖

职工	级别	工资
赵明	4	500
钱广	5	600
孙志	6	700
李开	5	600
周祥	6	700

问题根源：在属性如何取值的问题上，表结构与现实出现偏离



职工	级别
赵明	4
钱广	5
孙志	6
李开	5
周祥	6

级别	工资
4	500
5	600
6	700

关系规范化过程：从1NF到2NF

sno	sname	dno	dean	cno	grade
S01	杨明	D01	思齐	C01	90
S02	李婉	D01	思齐	C01	87
S01	杨明	D01	思齐	C02	92
S03	刘海	D02	述圣	C01	95
S04	安然	D02	述圣	C02	78
S05	乐天	D03	省身	C01	82

sno	sname	dno	dean
S01	杨明	D01	思齐
S02	李婉	D01	思齐
S03	刘海	D02	述圣
S04	安然	D02	述圣
S05	乐天	D03	省身

sno	cno	grade
S01	C01	90
S02	C01	87
S01	C02	92
S03	C01	95
S04	C02	78
S05	C01	82

关系规范化过程：从2NF到3NF

sno	sname	dno	dean
S01	杨明	D01	思齐
S02	李婉	D01	思齐
S03	刘海	D02	述圣
S04	安然	D02	述圣
S05	乐天	D03	省身

sno	sname	dno
S01	杨明	D01
S02	李婉	D01
S03	刘海	D02
S04	安然	D02
S05	乐天	D03

dno	dean
D01	思齐
D02	述圣
D03	省身

事务定义

事务是由一系列操作序列构成的执行单元，这些操作
要么都做，要么都不做，是一个不可分割的工作单位



all or none 要么全部 要么全不

事务ACID性质

原子性、一致性、隔离性、持久性

转账事务的定义

银行转帐：事务T从A帐户过户50¥到B帐户

read(A);

$A := A - 50;$

write(A);

read(B);

$B := B + 50;$

write(B);

set transaction isolation level read committed ;

start transaction;

update accounts **set** amounts = amounts - 50 **where** userId='A' ;

update accounts **set** amounts = amounts + 50 **where** userId='B' ;

commit;

并发控制：封锁的定义

- **封锁**就是一个事务对某个数据对象加锁，取得对它一定的控制，限制其它事务对该数据对象使用
- 要访问数据项 R ，事务 T_i 必须先申请对 R 的封锁，如果 R 已经被事务 T_j 加了不相容的锁，则 T_i 需要等待，直至 T_j 释放它的封锁



封锁的相容矩阵 $\text{comp}(A, B)$		
请求锁模式A	现有锁模式B	
	S	X
S	是	否
X	否	否

恢复控制：数据库备份

```
exec sp_addumpdevice 'disk', 'MyBKDB', DISK ='c:\MyBKDB.dat'
```

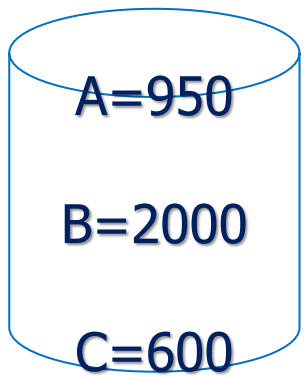
```
backup database LJCHEN to MyBKDB
```

```
restore database LJCHEN from MyBKDB
```


恢复控制：基于日志进行系统故障恢复

① 故障发生点， T_0 是成功事务

T_1 是失败事务



$\langle T_0, start \rangle$
 $\langle T_0, A, 1000, 950 \rangle$
 $\langle T_0, B, 2000, 2050 \rangle$
 $\langle T_0, commit \rangle$
 $\langle T_1, start \rangle$
 $\langle T_1, C, 700, 600 \rangle$

反向 *undo* T_1 操作

② *undo* $\langle T_1, C, 700, 600 \rangle$

A diagram of a database cylinder containing the value:
C=700

正向 *redo* T_0 操作

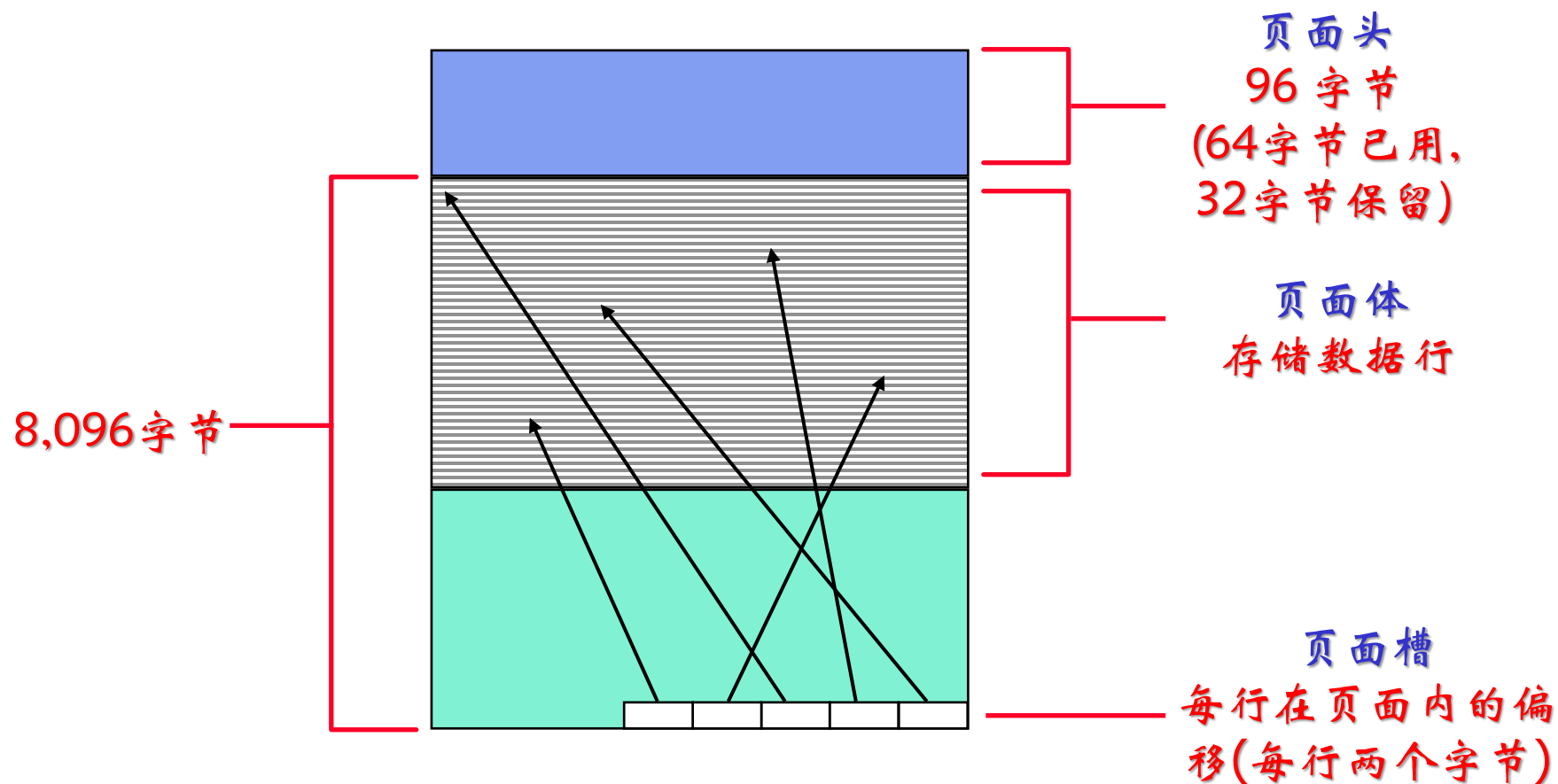
③ *redo* $\langle T_0, A, 1000, 950 \rangle$

④ *redo* $\langle T_0, B, 2000, 2050 \rangle$

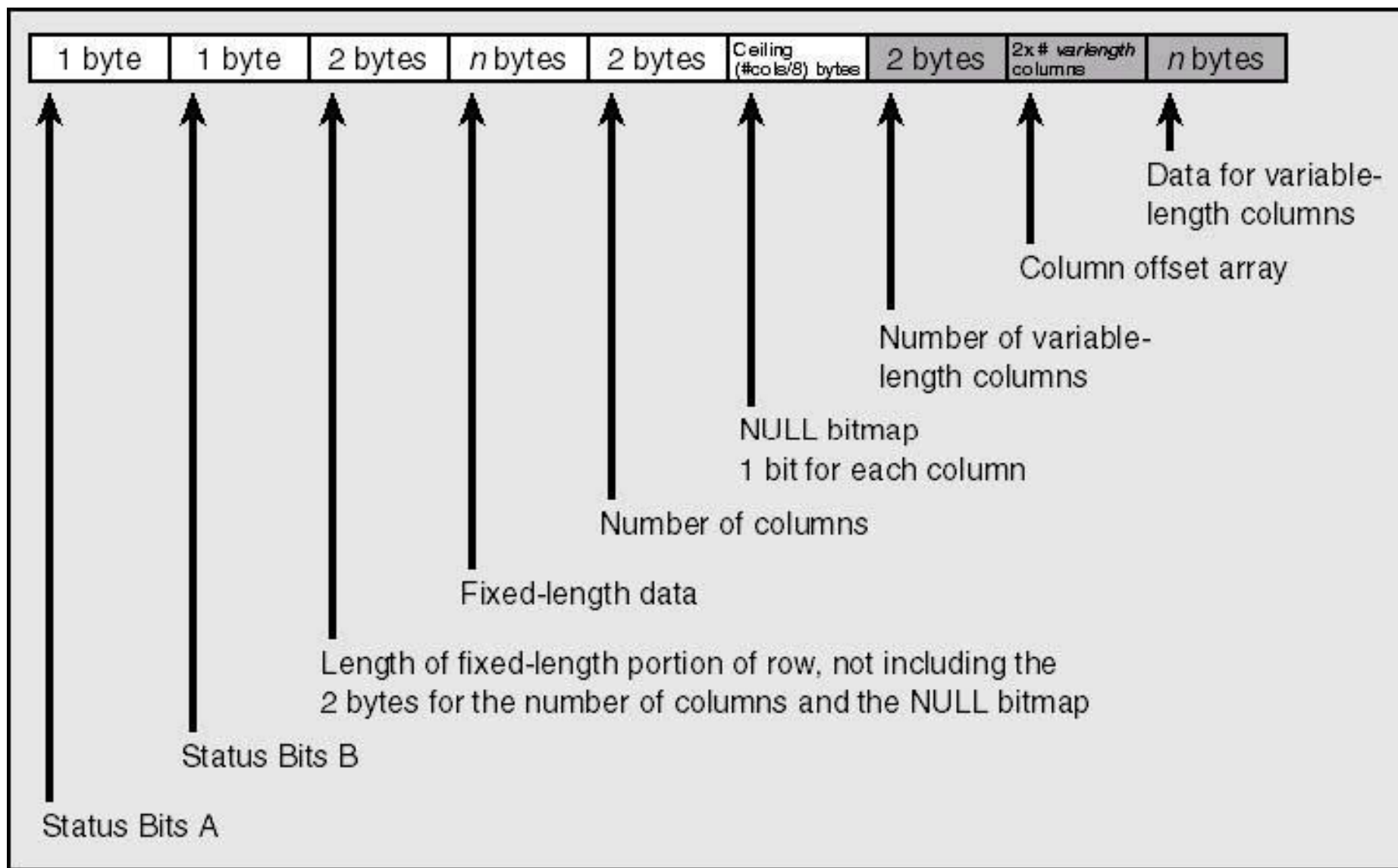
A diagram of a database cylinder containing the values:
A=950
B=2050
C=700

存储：基本页结构

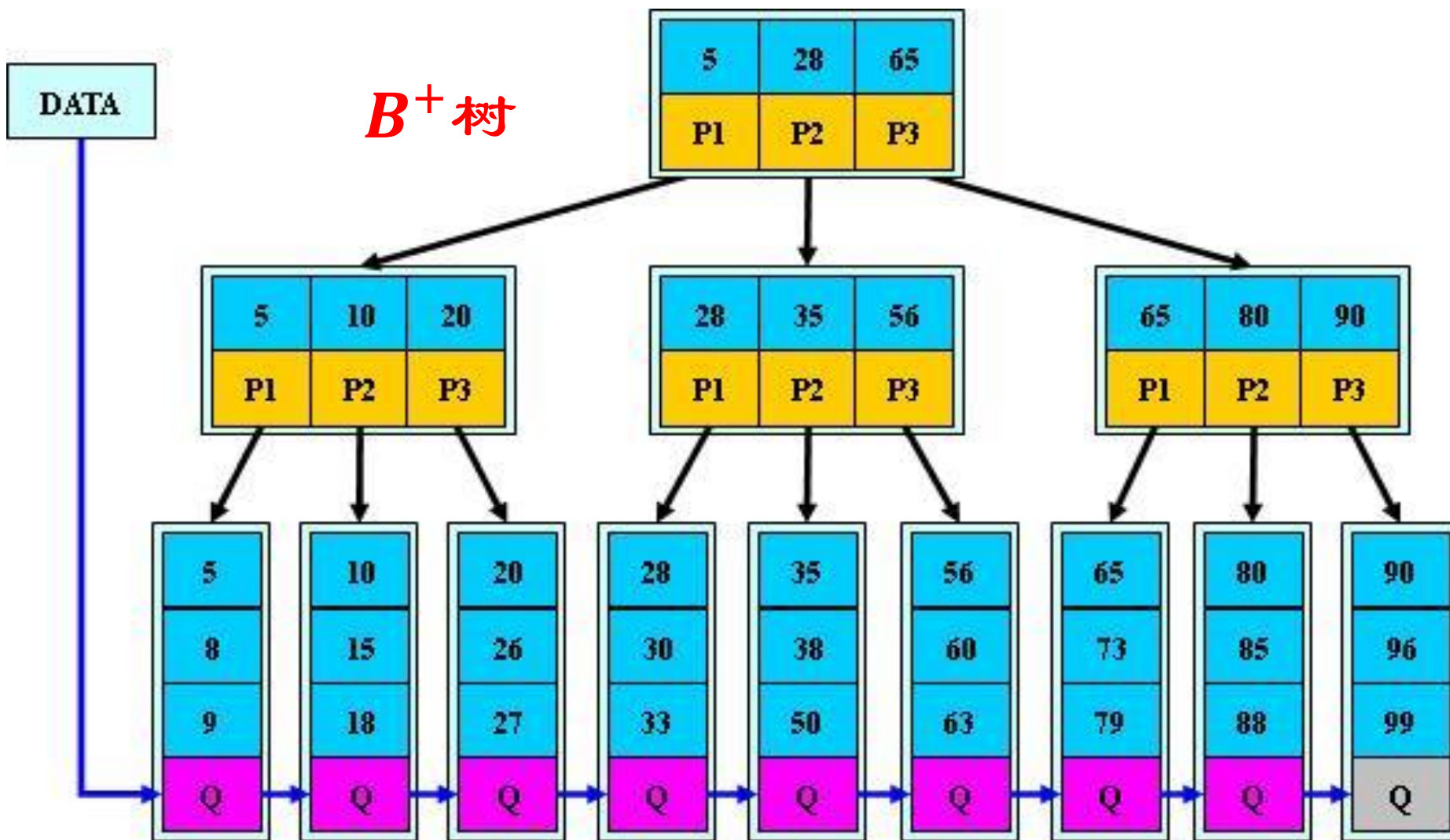
所有页面包括页面头、页面体、页面槽



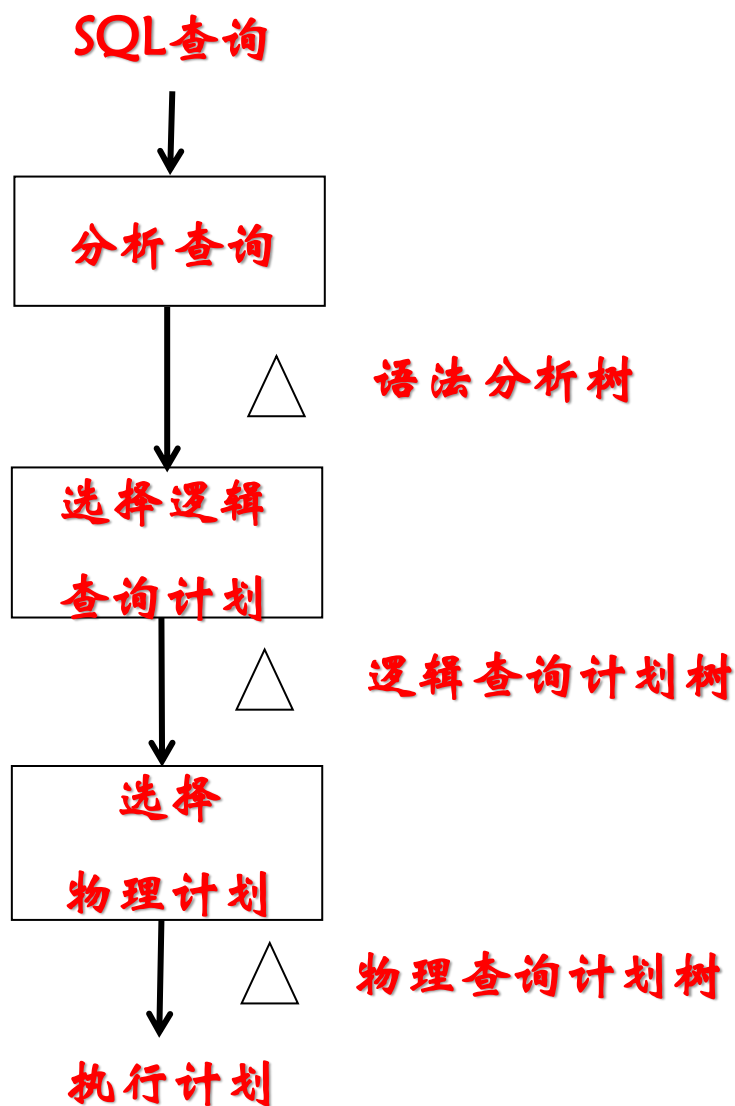
存储：数据行结构



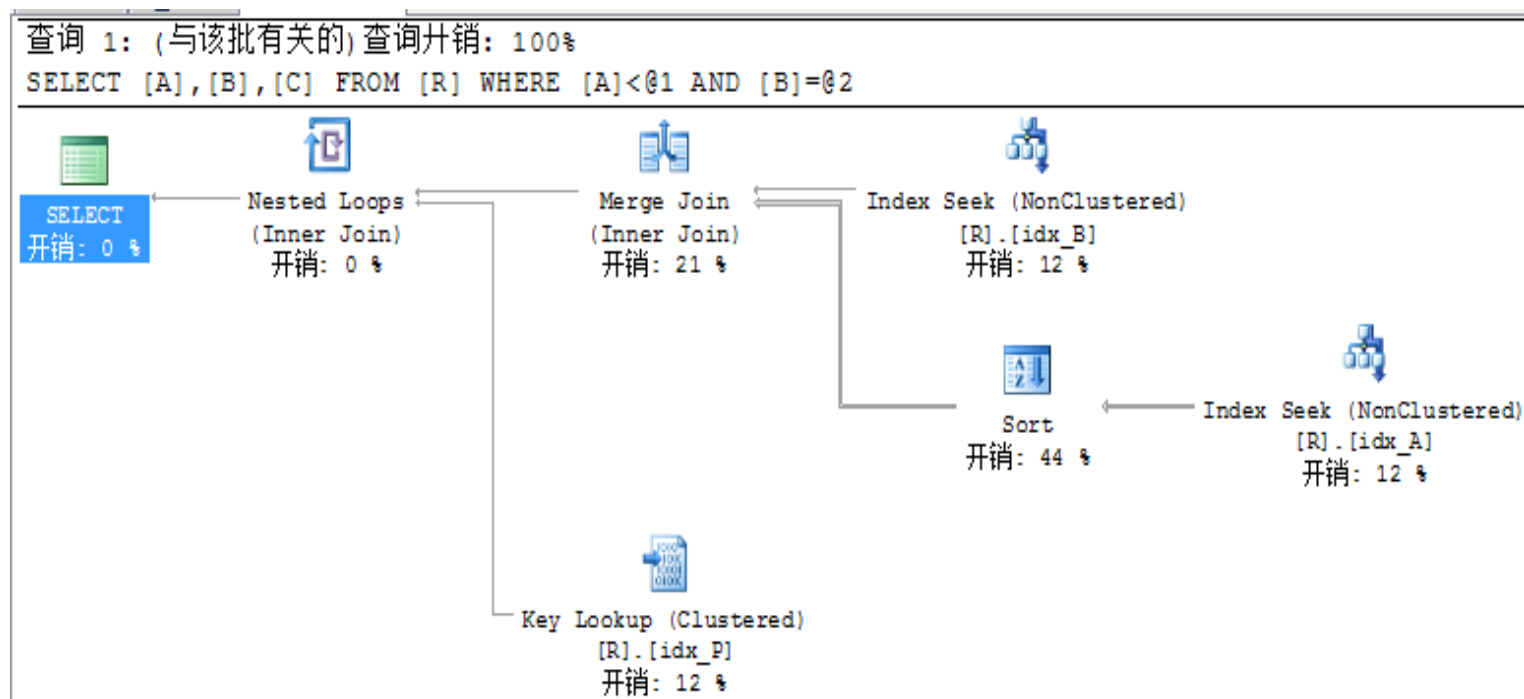
存储：索引结构



查询处理处理过程



```
create index idx_A on R(A) ,
create index idx_B on R(B)
select A, B, C from R
where A < 100 and B = 'C0000005794'
```



安全性控制

- 主体
- 客体
- 权限
- 加密
- 审计
- 视图
- SQL注入
- 统计数据库安全

```
grant select , insert on S to Liming  
with grant option  
revoke insert on S from Liming
```

SQL注入

```
SELECT * FROM users  
WHERE username = 'TomJerry'  
and (password = 'whoami' or 1=1)
```

完整性控制

- 主码约束：primary key
- 外码约束：foreign key
- 唯一性约束：unique
- 缺省值约束：default
- 空值约束：not null
- 条件约束：check
- 断言、触发器、函数...

```
create table emp(  
    eno          char(10) primary key,  
    ename        char(20),  
    mgr char(10) constraint FK_Constraint  
                foreign key references emp(eno)  
                deferrable initially immediate
```