



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

第十章 数据库测试

授课教师：郑炜

- 10.1 数据库应用软件测试
 - 10.1.1 数据库设计验证
 - 10.1.2 功能测试
 - 10.1.3 并发测试
 - 10.1.4 安全性测试
- 10.2 数据库管理系统基础测试
 - 10.2.1 数据库管理系统简介
 - 10.2.2 DBMS的SQL功能测试
 - 10.2.3 DBMS的事务特性测试

- 10.3 数据库管理系统性能测试
 - 10.3.1 DBMS性能测试的目的与性能指标
 - 10.3.2 DBMS 的基准性能测试
 - 10.3.3 DBMS 的性能测试工具
- 10.4 数据库管理系统高可用性测试

10.1.1 数据库设计验证



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY



OceanBase
A Powerful Database System



openGauss

MySQL®



PostgreSQL

数据库应用系统

数据库管理系统

10.1.1 数据库设计验证



■ Student (Sno, Sdept, Mname, Cname, Grade)

1. 数据冗余太大:

浪费大量的存储空间

例：每一个系主任的姓名重复出现

2. 修改复杂

数据冗余，更新数据时，维护数据完整性代价大。

例：某系更换系主任，系统必须修改与该系学生有关的每个元组

3. 插入异常

该插的数据无法插入到表中。

例：如果一个系刚成立，尚无学生，我们就无法把这个系及其系主任的信息存入数据库。

4. 删除异常

不该删除的数据被删掉了

例：如果某个系的学生全部毕业了，我们在删除该系学生信息的同时，把这个系及其系主任的信息也丢掉了

10.1.1 数据库设计验证



● 数据库设计步骤、成果与验证观点

编号	数据库设计步骤和输出成果	数据库设计验证观点
1	需求分析 输出：数据字典和数据流图	<ul style="list-style-type: none">- 是否反映所有的用户需求- 是否充分考虑系统的扩充和改变
2	概念结构设计 输出：ER 图 (Entity-Relation 图， 关系实体图)	<ul style="list-style-type: none">- 是否涵盖了系统涉及所有的实体和属性- 实体与属性的划分是否正确- 实体之间的联系与约束刻画是否准确全面- 不同子 ER 图中是否存在命名、结构等冲突- 不同子 ER 图中是否存在冗余
3	逻辑结构设计 输出：库表结构等	<ul style="list-style-type: none">- 是否符合关系数据库设计的范式理论，通常应达到 3NF 或者 BCNF- 是否考虑该系统的性能需求等进行去范式化- 是否考虑该系统的性能要求进行分库分表
4	物理结构设计 输出：存储、索引等	<ul style="list-style-type: none">- 索引设计(聚簇索引、唯一索引等)是否合理- 存储结构(关系、索引、日志，备份等)是否合理

10.1.2 功能测试



功能测试

通过测试验证软件的每个功能是否都按照用户的需求进行了实现并且能正常使用。

- 测试重点
 - 用户需求
 - WebApp, 移动App
- 测试方法
 - 白盒测试 + 黑盒测试

车次	出发站	到达站	出发时间	到达时间	历时	商务座	特等座	一等座	二等座	餐车	硬卧	软卧	硬座	站票	备注
G660	西安北	北京西	11:00	04:11	7	—	—	—	—	—	—	—	—	—	
G662	西安北	北京西	11:30	04:30	17	—	—	—	—	—	—	—	—	—	
G88	西安北	北京西	13:10	04:32	—	—	—	—	—	—	—	—	—	—	
K1364	西安	北京西	14:11	15:44	—	—	—	—	—	—	—	—	—	—	
G664	西安北	北京西	14:34	04:54	—	—	—	—	—	—	—	—	—	—	
G666	西安北	北京西	16:00	04:51	—	—	—	—	—	—	—	—	—	—	
G668	西安北	北京西	17:00	04:49	—	—	—	—	—	—	—	—	—	—	
T232	西安	北京西	17:44	12:44	—	—	—	—	—	—	—	—	—	—	
G670	西安北	北京西	18:05	04:23	—	—	—	—	—	—	—	—	—	—	
Z44	西安	北京西	19:15	13:11	—	—	—	—	—	—	—	—	—	—	
Z20	西安	北京西	19:30	11:41	—	—	—	—	—	—	—	—	—	—	
T42	西安	北京西	19:36	13:46	—	—	—	—	—	—	—	—	—	—	
T36	西安	北京西	20:17	10:15	—	—	—	—	—	—	—	—	—	—	
T8	西安	北京西	22:47	13:41	—	—	—	—	—	—	—	—	—	—	

性能测试:

验证系统的主要功能或组件的实际性能是否与用户的性能需求一致。

数据库应用软件的性能测试:

关注该软件整体或者其中某些组件在规定时间内，响应用户或系统输入的能力，通常用请求响应时间，吞吐量等度量性能优劣。

常见工具: LoadRunner、Jmeter、Locust

- 负载测试 (Load Test)

通过逐渐增加负载评估系统或者组件性能的测试方法，目的是观察系统在各种不同的负载情况下是否都能够正常工作。而性能测试通常是直接指定一个特定的负载进行测试。

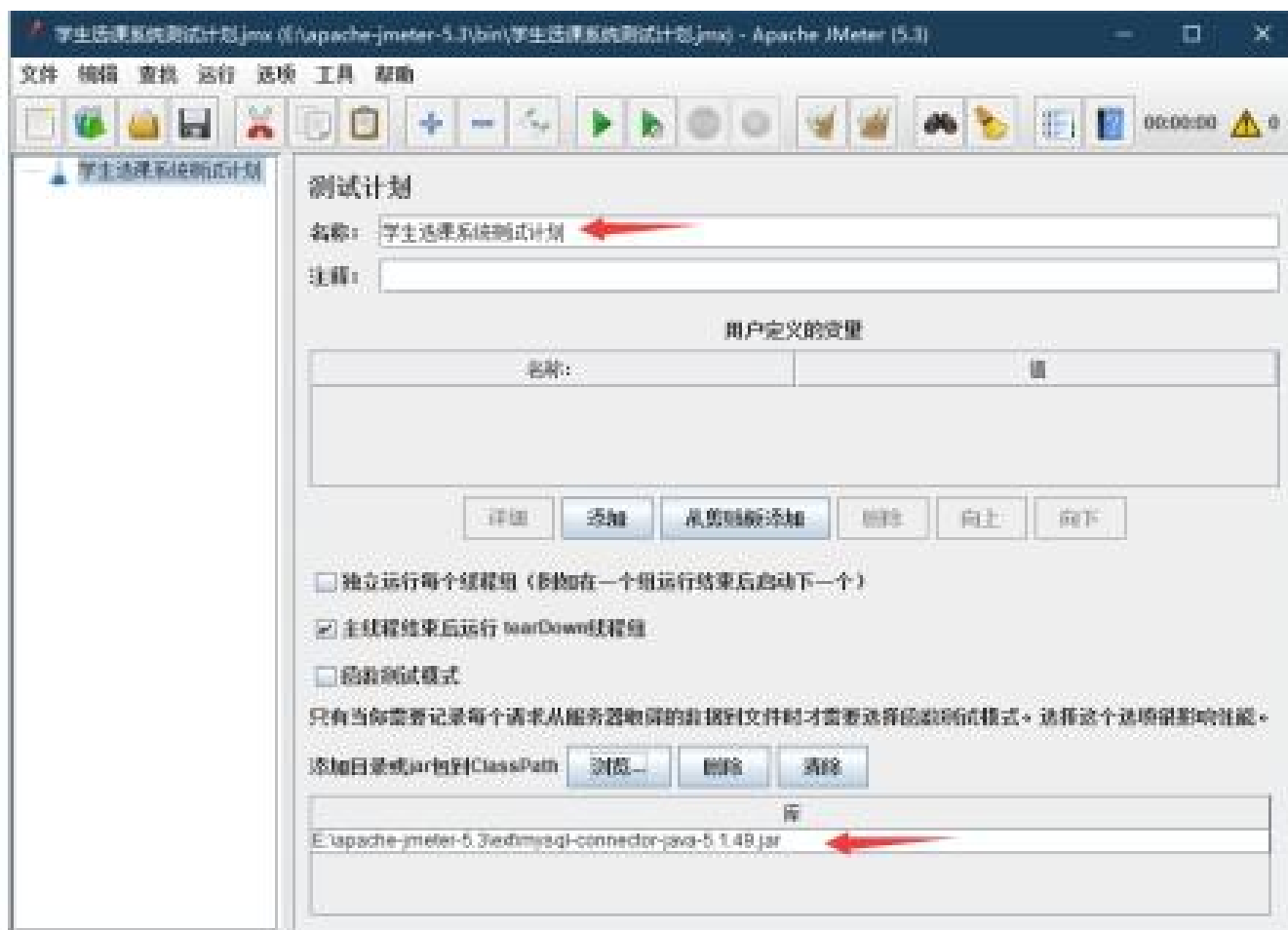
- 压力测试 (Stress Test)

评估系统或者组件处于或超过预期负载时的运行情况。重点关注系统在峰值负载或超出最大负载情况下的处理能力，在压力级别逐渐增加时，系统性能应该是按照预期缓慢下降，但是不应发生系统崩溃的现象。

10.1.3 性能测试



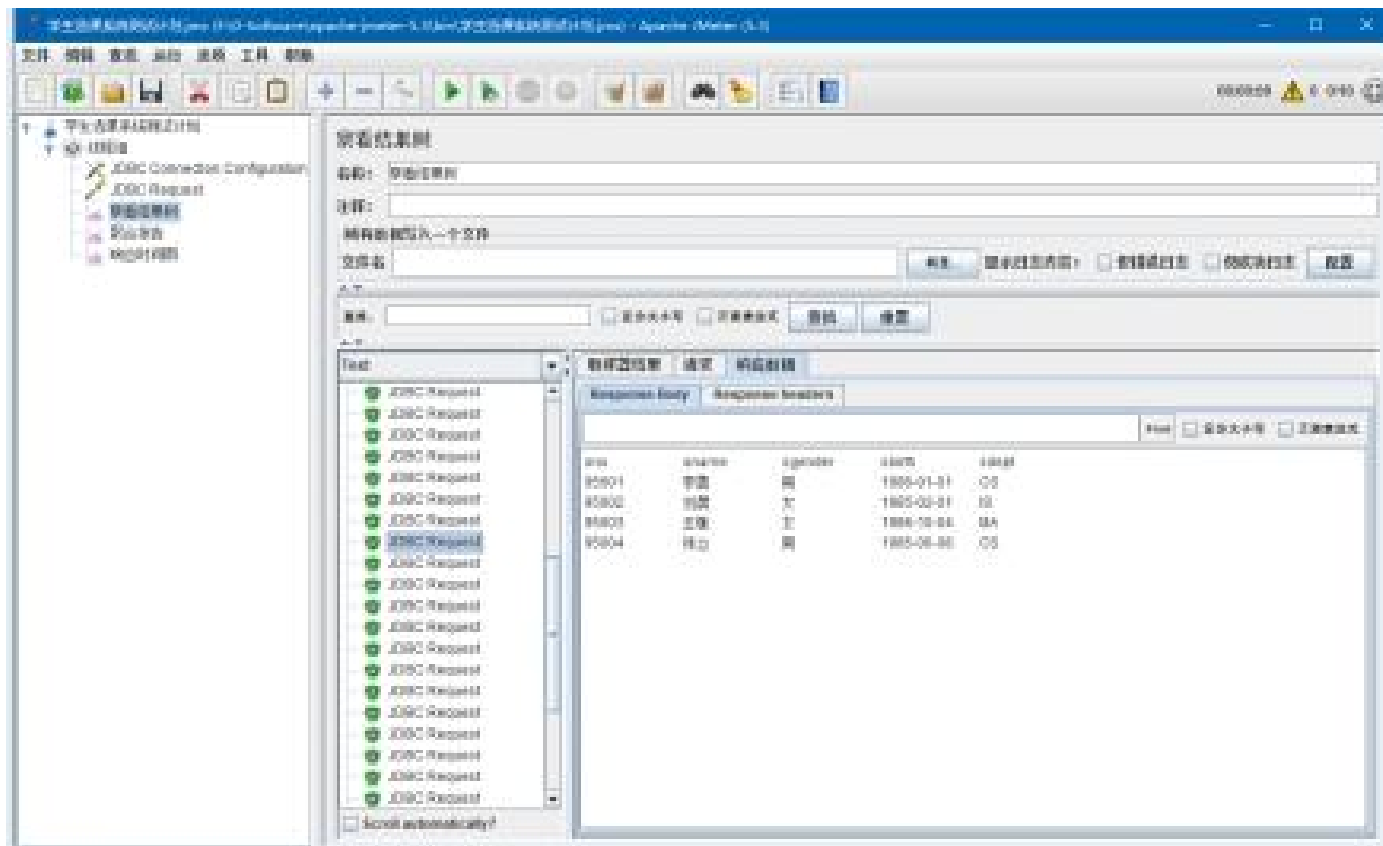
基于Jmeter的测试示例：



10.1.3 性能测试



基于Jmeter的测试示例：



• SQL注入

```
select * from member where UID = ' "& request("ID")  
&" ' and Passwd = ' "& request("Pwd") ' '
```



```
select * from member where UID = ' Admin ' -- ' And  
Passwd = ' '
```

假设猜测系统有Admin的用户，构造以上SQL语句，由于SQL语言中 “--” 是注释符号，其后的字符都会被当作注解，因此该攻击者无需输入密码可直接进入系统



- 针对SQL注入的测试

- 构造专门的注入字符或者语句单独测试:

- 例如给可输入的用户名构造 `zhang'`

- 借助漏洞扫描工具测试:

- 扫描之后, 进行对应的升级数据库等动作



- 10.1 数据库应用软件测试
- 10.2 数据库管理系统基础测试
 - 10.2.1 数据库管理系统简介
 - 10.2.2 DBMS的SQL功能测试
 - 10.2.3 DBMS的事务特性测试

● 数据库管理系统 (Database Management System, DBMS)

是一种控制和管理数据库的大型基础软件，用于建立、使用和维护数据库。

- 数据定义功能：库、表、视图、索引等
- 数据操控功能：数据的增删改查
- 数据库运行管理功能：完整性、安全性等
- 数据库的维护功能



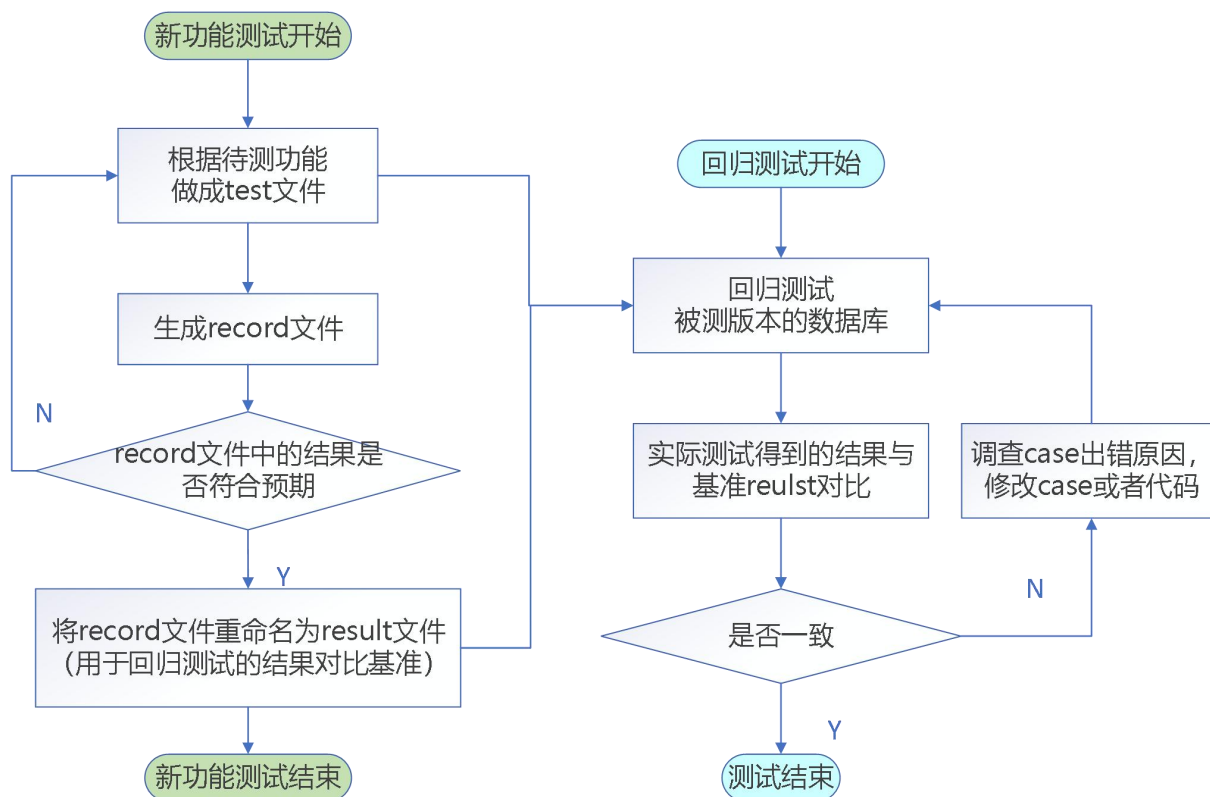
openGauss



10.2.2 DBMS的SQL功能测试



- 手动测试：根据SQL手册，设计测试用例并手动执行SQL。
- 自动测试：MySQLTest自动化测试框架



10.2.2 DBMS的SQL功能测试



测试用例文件（每个.test是一个测试用例）

Sql > mysql-8.0.22-source > mysql-8.0.22 > mysql-test > t				在 t 中搜索
名称	修改日期	类型	大小	
1st.test	2020/9/23 20:37	TEST 文件	1 KB	
admin_interface.test	2020/9/23 20:37	TEST 文件	13 KB	
admin_interface_ipv4_mapped.test	2020/9/23 20:37	TEST 文件	2 KB	
alias.test	2020/9/23 20:37	TEST 文件	10 KB	
all_persisted_variables.test	2020/9/23 20:37	TEST 文件	9 KB	



测试用例基准结果（.result与测试用例逐一对应）

MySQL > mysql-8.0.22-source > mysql-8.0.22 > mysql-test > r				在 r 中搜索
名称	修改日期	类型	大小	
1st.result	2020/9/23 20:37	RESULT 文件	1 KB	
admin_interface.result	2020/9/23 20:37	RESULT 文件	9 KB	
admin_interface_ipv4_mapped.result	2020/9/23 20:37	RESULT 文件	1 KB	
alias.result	2020/9/23 20:37	RESULT 文件	15 KB	
all_persisted_variables.result	2020/9/23 20:37	RESULT 文件	3 KB	
almost_full.result	2020/9/23 20:37	RESULT 文件	2 KB	

Test文件内容示例

```
1 # InnoDB page size 16k is required
2
3 #
4 # Test of alter table
5 #
6
7 SET sql_mode = 'NO_ENGINE_SUBSTITUTION';
8
9 SET SESSION information_schema_stats_expiry=0;
10
11 create table t1 (
12   col1 int not null auto_increment primary key,
13   col2 varchar(30) not null,
14   col3 varchar (20) not null,
15   col4 varchar(4) not null,
16   col5 enum('PENDING', 'ACTIVE', 'DISABLED') not null,
17   col6 int not null, to_be_deleted int);
18 insert into t1 values (2,4,3,5,"PENDING",1,7);
19 alter table t1
20   add column col4_5 varchar(20) not null after col4,
21   add column col7 varchar(30) not null after col5,
22   add column col8 datetime not null, drop column to_be_deleted,
23   change column col2 fourth varchar(30) not null after col3,
24   modify column col6 int not null first;
25 select * from t1;
26 drop table t1;
```

MySQLTest语法参考：

https://dev.mysql.com/doc/dev/mysqldb/latest/PAGE_MYSQL_TEST_RUN.html



- 自动测试:

MySQLTest自动化测试框架事务（数据操作的最小逻辑单元）特性：ACID

- 原子性(Atomic): 该事物中的SQL语句或者全部执行，或者全部不执行。
- 一致性(Consistent): 一个事务在执行之前和执行之后，数据库的数据都必须处于一致的状态
- 隔离性(Isolation): 在并发环境中，并发的事务时相互隔离的，一个事务的执行不能被其他事务干扰。
- 持久性(Duration): 一旦事务提交，事物对数据库中的对应数据的状态变更就会永久保存到数据库中。

●数据不一致问题与隔离级别

隔离级别	脏读	不可重复读	幻读
读未提交(Read uncommitted)	允许	允许	允许
读已提交(Read committed)	不允许	允许	允许
可重复读(Repeatable read)	不允许	不允许	允许
可串行化(Serializable)	不允许	不允许	不允许

10.2.3 DBMS的事务特性测试



- 测试方法：同时开启多个客户端连接，设置不同隔离级别，测试不同的并发操作。

```
1 * set transaction isolation level read uncommitted;
2 * set autocommit=0;
3 * select * from sj;
4 * update s set sdept = 'IS' where sno='95004';
```

sno	sname	sgender	sbirth	sdept
95001	李勇	男	1986-01-01	CS
95002	刘晨	女	1985-02-01	IS
95003	王敏	女	1986-10-04	MA
95004	张立	男	1985-06-08	CS

```
1 * select * from sj;
```

sno	sname	sgender	sbirth	sdept
95001	李勇	男	1986-01-01	CS
95002	刘晨	女	1985-02-01	IS
95003	王敏	女	1986-10-04	MA
95004	张立	男	1985-06-08	IS

● 原子性

● 持久性

右侧操作结束后，退出再次进入，数据已经正常保存

```
create table test_11 (c int primary key,d int);
```

场景 1 全部回滚

```
Begin work;
```

```
Insert into test_11 values(1,1); --执行成功
```

```
Insert into test_11 values(2,2); --执行成功
```

```
Rollback; --执行成功
```

```
Begin work;
```

```
select * from test_11 ; --执行成功，返回空结果集
```

```
commit;
```

场景 2 全部提交

```
Begin work;
```

```
Insert into test_11 values(1,1); --执行成功
```

```
Insert into test_11 values(3,1); --执行成功
```

```
commit;
```

```
Begin work;
```

```
select * from test_11 ; --执行成功，返回记录为： (1,1) 和 (3,1)
```

```
commit;
```

10.2.3 DBMS的事务特性测试



● 隔离性

用户 1	用户 2
SQL>begin work;	SQL>begin work;
SQL>insert into test_1 values(3,1); —插入一条语句	SQL>insert into test_1 values(3,33); —插入一条语句
SQL>select * from test_1 where a=3; A B _____ 3 1 —查询出一条数据, 无法查询出用户 2 插入数据	SQL>select * from test_1 where a=3; A B _____ 3 33 —查询出一条数据, 无法查询出用户 1 插入 数据
SQL>commit;	SQL>commit;
SQL>select * from test_1 where a=3; A B _____ 3 1 3 33	SQL>select * from test_1 where a=3; A B _____ 3 1 3 33

10.2.3 DBMS的事务特性测试



● 一致性

用户 1	用户 2
SQL>begin work; SQL>Insert into test values(1,11,'test1');	
	SQL>select * from test; —查出 0 条数
SQL>commit; — SQL operation complete.	
	SQL>select * from test; A B C ——— ——— ——— 1 11 test1 — 查询出一条记录



- 10.1 数据库应用软件测试
- 10.2 数据库管理系统基础测试
- 10.3 数据库管理系统性能测试
 - 10.3.1 DBMS性能测试的目的与性能指标
 - 10.3.2 DBMS 的基准性能测试
 - 10.3.3 DBMS 的性能测试工具



- **DBMS性能测试的目的：** 为了验证系统是否能满足到用户提出的性能指标，发现性能瓶颈，优化系统整体性能。

常见的DBMS性能度量指标：

1. 每秒执行事务数TPS (Transactions Per Second)

数据库每秒执行的事务数，以COMMIT成功次数为准。

2. 每秒执行请求数QPS (Queries Per Second)

数据库每秒执行的SQL数。

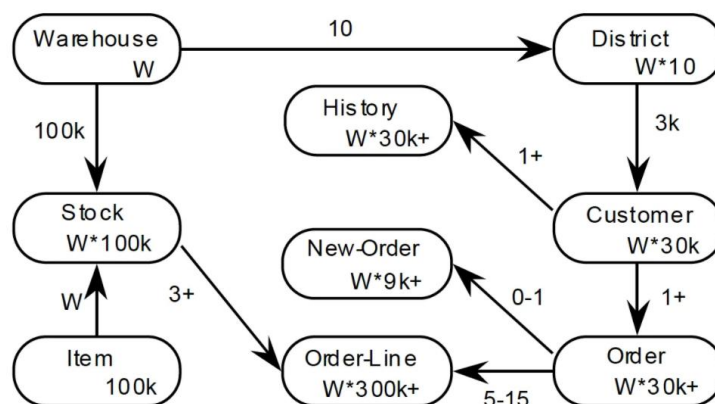
3. 响应时间RT(Response Time)

数据库执行每个请求的响应时间。

注意：每个测试工具都有自己的度量，但总体是度量单位时间系统的处理能力

● DBMS的基准性能测试

- OLTP (Online Transaction Processing) 联机业务处理测试标准: **TPC-C**、TPC-E (最新)
- OLAP (Online Analysis Processing) 决策支持/大数据测试标准: **TPC-H**、TPC-DS (最新)



● 常用工具：Sysbench、BechmarkSQL(TPC-C)等

Sysbench主要的OLTP测试SQL语句

```
mysql> show tables;
+-----+
| Tables_in_sbtest |
+-----+
| sbtest1           |
| sbtest2           |
| sbtest3           |
+-----+
3 rows in set (0.00 sec)

mysql> select count(*) from sbtest1;
+-----+
| count(*) |
+-----+
| 10000    |
+-----+
1 row in set (0.09 sec)
```

测试场景	测试观点	用例分类	数量	SQL 语句
OLTP 只读	主键点查询	point_select	10	SELECT c FROM sbtest%u WHERE id=?
	范围查询	simple_ranges	1	SELECT c FROM sbtest%u WHERE id BETWEEN ? AND ?
		sum_ranges	1	SELECT SUM(k) FROM sbtest%u WHERE id BETWEEN ? AND ?
		order_ranges	1	SELECT c FROM sbtest%u WHERE id BETWEEN ? AND ? ORDER BY c
		distinct_ranges	1	SELECT DISTINCT c FROM sbtest%u WHERE id BETWEEN ? AND ? ORDER BY c
OLTP 只写	增删改	index_update	1	UPDATE sbtest%u SET k=k+1 WHERE id=?
		non_index_update	1	UPDATE sbtest%u SET c=? WHERE id=?
		delete	1	DELETE FROM sbtest%u WHERE id=?
		insert	1	INSERT INTO sbtest%u (id, k, c, pad) VALUES (?, ?, ?, ?)
OLTP 混合 读写	OLTP 只读	-	14	同 OLTP 只读
	OLTP 只写	-	4	同 OLTP 只写



Sysbench: <https://github.com/akopytov/sysbench/>

主要测试命令:

- `sysbench oltp_read_write.lua --tables=3 --table_size=10000 --mysql-user=root --mysql-password=123456 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-db=sbtest prepare`
- `sysbench oltp_point_select.lua --tables=3 --table_size=10000 --mysql-user=root --mysql-password=123456 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-db=sbtest --threads=100 --time=100 --report-interval=5 run`
- `sysbench oltp_point_select.lua --tables=3 --table_size=10000 --mysql-user=root --mysql-password=123456 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-db=sbtest --threads=100 --time=100 --report-interval=5 cleanup`

10.3.3 DBMS 的性能测试工具



Sysbench 测试结果

```
[root@localhost lua]# sysbench oltp_point_select.lua --tables=3 --table_size=10000 --mysql-user=root --mysql-password=123456 --mysql-port=3306 --mysql-db=test --threads=100 --time=100 --report-interval=5 run
sysbench 1.1.0-797e4c8 (using bundled LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 100
Report intermediate results every 5 second(s)
Initializing random number generator from current time

Initializing worker threads...

Threads started!

[ 5s ] tidx: 100 tps: 12918.20 qps: 12918.20 (r/w/o: 12918.20/0.00/0.00) lat (ms,95%): 13.70 err/s: 0.00 reconf/s: 0.00
[ 10s ] tidx: 100 tps: 13307.71 qps: 13307.71 (r/w/o: 13307.71/0.00/0.00) lat (ms,95%): 13.22 err/s: 0.00 reconf/s: 0.00
[ 15s ] tidx: 100 tps: 13279.14 qps: 13279.14 (r/w/o: 13279.14/0.00/0.00) lat (ms,95%): 12.98 err/s: 0.00 reconf/s: 0.00
[ 20s ] tidx: 100 tps: 13429.99 qps: 13429.99 (r/w/o: 13429.99/0.00/0.00) lat (ms,95%): 13.32 err/s: 0.00 reconf/s: 0.00
[ 25s ] tidx: 100 tps: 12887.88 qps: 12887.88 (r/w/o: 12887.88/0.00/0.00) lat (ms,95%): 13.46 err/s: 0.00 reconf/s: 0.00
[ 30s ] tidx: 100 tps: 13499.96 qps: 13499.96 (r/w/o: 13499.96/0.00/0.00) lat (ms,95%): 12.98 err/s: 0.00 reconf/s: 0.00
```

```
SQL statistics:
  queries performed:
    read: 1287596
    write: 0
    other: 0
    total: 1287596
  transactions: 1287596 (12871.28 per sec.)
  queries: 1287596 (12871.28 per sec.)
  ignored errors: 0 (0.00 per sec.)
  reconnects: 0 (0.00 per sec.)

Throughput:
  events/s (eps): 12871.2772
  time elapsed: 100.0364s
  total number of events: 1287596

Latency (ms):
  min: 0.05
  avg: 7.77
  max: 2401.94
  95th percentile: 12.30
  sum: 10000767.70

Threads fairness:
  events (avg/stddev): 12875.9600/128.20
  execution time (avg/stddev): 100.0077/0.01
```

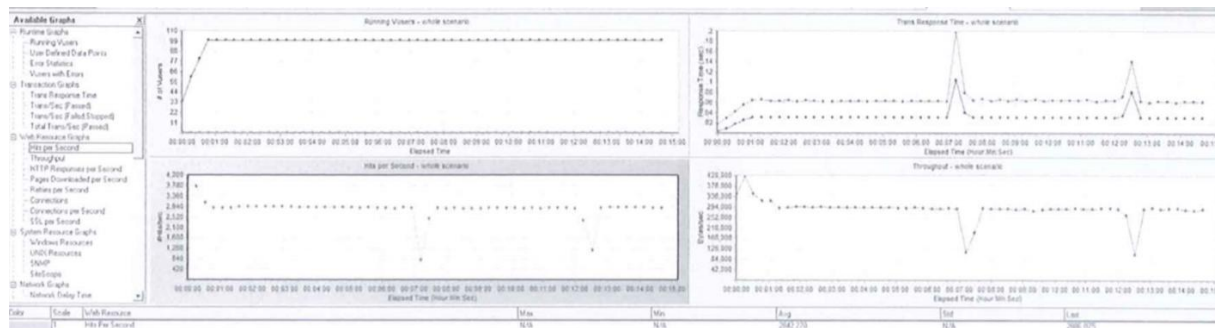



- 10.1 数据库应用软件测试
- 10.2 数据库管理系统基础测试
- 10.3 数据库管理系统性能测试
- 10.4 数据库管理系统高可用性测试

- DBMS的高可用性测试
- 高可用：指在DBMS发生故障时恢复业务运行的能力。
- 通常采用高可用架构：例如一主N备；三地两中心等
- 高可用测试：通过故障模拟（混沌测试），例如利用工具Jepsen、ChaosBlade等模拟网络故障，CPU超载，服务器宕机，内存不足等，测试在这些极端条件下系统的处理。



- DBMS的高可用性测试
- 模拟客户端：Jmeter / LoadRunner等，持续发送并发请求给DBMS（包含读写操作或者特定业务）
- 故障模拟工具
- 观察结果：故障发生时，可以快速切换到备节点提供服务，服务能力可以有所下降，但是不应该出错。



- 数据库应用软件测试

数据库设计验证、功能测试、性能测试

- 数据库管理系统基础测试

功能测试(MySQLTest)、事务测试、安全性测试

- 数据库管理系统性能测试

TPC-C、TPC-H等

Sysbench、 BenchmarkSQL

- 数据库管理系统高可用性测试（混沌测试）



THANK YOU

The End

