Date :

→ SQL is pronounced as 'si:kwəl' (Sequel)

→ if we follow best practices in designing a D.B. most of testing efforts are reduced.

→ every table in RDBMS should be a primary key.

→ modified Date should be system Date.

Databases
R.C → New D.B. (Emp)

menu bar → New query.
D.B → New query.

Emp
↳ Tables
R.C Table...

→ To check constraints / str / schema etc of table.

Table
R.C → script table as → { Create To → New Query Editor Win
Drop To
Insert
Delet
Update
Select

D.B. Schema : a skeleton str-rep
how data is organized & how rel^n are associated.
formulates all constraints

13-11-2019

D.B. Testing

— By Tousif Borkar.

Date :

MON TUE WED THU FRI SAT SUN

D.B. organized coll^n of Data.

NoSQL D.B.MS
- mongoDB
- Cassandra

DBMS → software used to store & manage data
→ MS Access → No rel^n ship
→ Redundancy, support single user.

(There is no rel^n b/w tables in DBMS)

RDBMS → SQL S^r, mySQL, Oracle, PostgreSQL
→ Advance version of DBMS

Prerequisite of D.B.Tg
- D.B. Construct
- Stored Procedure
- Triggers
- Join

OLAP → Analytical (Datawarehouse) → Dimensional model
- facts (numbers)
- Dim table
- facts Table

OLTProcessing → Transactional → Relational model
→ Normaliz^n is promoted in OLTP but not in OLAP system, support denormalized table

→ prim key with single col is preferred → can be meaningless to end user. not composite col.

AUTH: SQL S^r Auth^n
- sa
- Password_123

10.30.5.138

windows Authentic^n for Local instance.

Characteristics of DBMS
→ Real-world entity : Banking Sy. Hosp. Sy.
→ Rel^n based table
→ Acid Properties
- → Atomicity — Txn property. All or nothing (Locking Technique)
- → Consistency
- → Isolation
- → Durability.

→ Security

D.B
- → Structure (Schema Tg) → Structural Tg
- → Data → fun^l Tg (→ Data correct ness e.g. truncation of value. → payment gateway (Banks))
- → (Performance, Security Tg) Non-fun^l Tg.

Layers in App^n

Client L^r → App^n L^r → Data L^r.

- Client L^r: (GUI) multi L^r (mob. web)
- App^n L^r: (Business logics (multi-layers))
- Data L^r: (Data storage) change in DB via stored procedure
  - every statement is converted into a txn.

Import a D.B.
Restore.

1 Tier App^n: all 3 L^r at one place — MS office

2 Tier App^n: one L^r for Data, Client & other for Business lay & Data logic

3T^r: are 3 L^r kept separately.

~~Databases~~
└ RestoreDB
  └ Device
    └ Add
      └ selectPath.

Table	col	comment
Dept	depId	P.Key should be autogenerated
	DepNm	should be NotNull
Employee	empId	P.Key should be autogenerated, should be INT gr than 0.

Given By client for Testing

Table Name	ColName	DataType	Length	Nullable	Key	B.Rules	Comm
production.Product.	ProductId	varchar	20	NO	PK	> 0	
	modifiedDate	date		No		SystemDate	
		Bigint		Yes	F.K		

Mapping Sheet

Date :

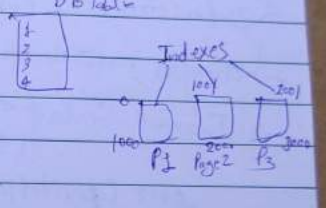
~~online processing System~~ :

OLAP → ~~online DB query Answering system~~, more processing Time

OLTP → online database modifying System. Pro. Time is less.

used in D.B performance Tg.

Indexes → clustered → uses P. Key, ~~physically~~ sorted.
→ Non-clustered Index.

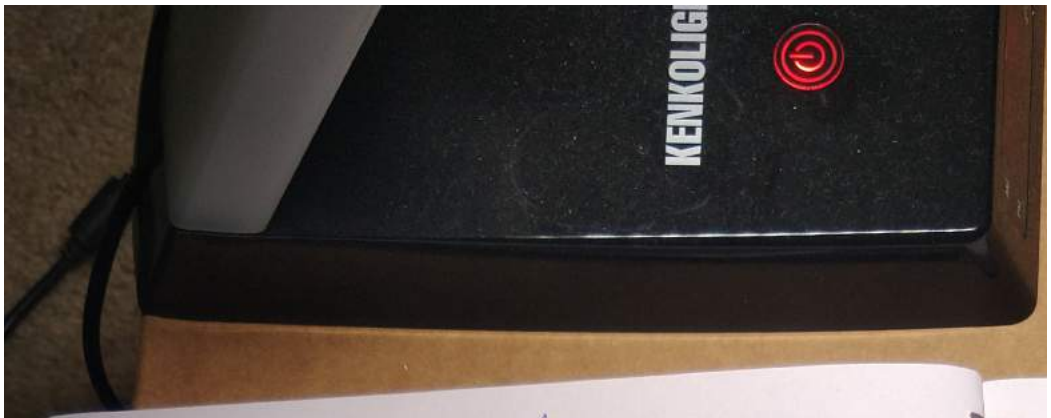


Query Plan → how Indexes are working
→ graphical rep^n of index working

Subquery vs Joins : Same thing can be achieved using both.
join is better as query plan is created by DB SQL engine & same so next execut^n faster.

Any ~~change in~~ query will result in new Q plan Creation.

Tg of constraints, store proc, triggers is considered if D.B. is complex.



Triggers → Before or After } Oracle.
event

INS
DEL
UP → DB.

SP → roll back to avoid Del".

exec SP

D.B. Tg challenges.

→ Large scope of Tg
Amount of Data
Data Correctness (-ve Age, ABC Name)

→ changes in D.B. schema & data (Analyze impact / modify impated T.C. / write New Tg if needed / Test again)

→ Incorrect Scaled-down D.B.
(No prod like D.B.)
Tester dev D.B. have little Data

→ Lack of skills: T^rs should be comfortable with SQL query & required DB Tool.

over (Partition By clause
Group By) Alternative

Select ISNULL(Quntity, 0), Name from Table

→ if Qty is NULL show it as 0

//Resolvetx / f" / f" | Trigger content | D.B Tg

↕

→ Trig | Trig CTI | TD ISLO | Database Tg

(@ID int,
@FName varchar(20),

SQL's Authentic"
{ Sa
{ Password_123

execute proc 1, 'John', 'POT', 'male'

Begin

Begin Txn
Begin Try

check for proc"
→ there should be proper exc" handling in proc" (try-catch)
→ correct Data types are used for col names

end

Date :

sp_help 'patient_dbo' → To see structure of patient table

D.B.

Testcases Template

TCName	Desc^n / stepName	Desc^n steps	Expected	Actual	Type	Status (P/F)	Design
TC01_PatientTableStr	Verify datatype & constraint of patient table as per mapping sheet.	1. 2.	constraints should match with M.S.	should match with M.S.	manual		

Structural Tg:

→ Verify Table str (cl name, DataType, length) constrain etc.

→ Column Name should be relevant

→ Verify all the columns are present.

Functional Tg.

Insert → Verify Values inserted from GUI is inserted in respective fields.

Update → Update from UI verify update in backend

→ check all constraints including NULL value.

Delete → check if delete op^n works.

Neg Scenario

Insert same value in backend directly & check its display on UI

Date :
MON TUE WED THU FRI SAT SUN

Use DBTestPortal10

Delete from PatientAddress;
Delete from PatientDetails;

Normaliz^{n}: → technique to organise ~~tables~~ in a manner that reduces redundency & dependency of data.
→ It divides long table into S^{l} Tab. & links them using relationships.

1NF → Each cell in table for a col shud contain single value
2NF → store data in multi table to avoid redundancy.
3NF → Removing transitive dep. in table of diff col
↓ sufficient.

tinyint → 0-255

⇒ Schema check in heterogeneous D.B. → migrat^{n}.

non-clustered index for region, gender

→ Truncation Issues in Table.

→ Should be able to apply join among 5-6 tables.

Why D.B.Tg
↳ To increases the robustness of data.
↳ To validate ACID properties.
Data mapping, Integrity, B.Rule conformance

Types of D.B.Tg ① Structured Tg (schema Tg, mapping Tg DI for BE
col Nam, D.T., length, constraints (P.K. F.K.
Naming conv^{n}.
② F^{t} Tg
③ Nonfunct^{l} Tg ← Perf^{n} → Load Tg → Volume Tg / F.load Tg
Security. (Data encryp^{n} - stored data)

valid^{n}
{ Mandatory fields: No nulls.
Rules valid^{n} :- whether rules specified in mapping are correct in coding?
Naming conv^{n}

{ Data Accuracy. ~~story as per req.~~ commit, roll back → No data shud be saved for failed Txn.
→ ACID

D.B Tg: Tg can be done at all 3 L^rs.

~~Usually consist of layered process.~~

Client, Business, Data L^r.

DB Tg Includes performing:

Data validity

data integrity Tg

performance check (Query exe^n time)

Tg of Procedures, triggers, f^ns in the D.B.

Following verif^n are carried out during D.B Tg.

→ checking Data mapping [UI ⇄ DB] (data complete & correct in sync with source (e.g UI))

→ ACID property valid^n (No truncation, No Additional data inserted, 1 Txn may send data into multiple tables.)

→ Data Integrity (Accuracy of data, it includes data valid^n before/After insert^n, updates, & del^n.

→ Business rule conformance.

→ Tg DB includes review E/R (entity rel^nship) Diagram.

D.B. design

D.B objects: Table, views, stored procedured etc. indexes

→ validate schema, tables, columns, Keys & indexes

↓

Name Datatype, length ~~Keys~~ constraints.

→ Check Not NULL (mandatory) field carefully.

→ There should be Audit log for every CRUD oper^n

→ Name length in D.B. should display prop. on UI

No trunc^n on UI

→ Test Stored proc^r & triggers with sample i/p data.

→ ~~checklist for D.B. Tg~~ → PPT.