



# MySQL - RDBMS

Trainer: Mr. Nilesh Ghule



# SQL Keys ↩

- 🔑 An SQL key is either a single column (or attribute) or a group of columns that can uniquely identify rows (or tuples) in a table.
- 🔑 Super key is a single key or a group of multiple keys that can uniquely identify tuples in a table.
- 🔑 Candidate key is a single key or a group of multiple keys that uniquely identify rows in a table.
- 🔑 Primary key is the Candidate key selected by the database administrator to uniquely identify tuples in a table. (designer)
- 🔑 Alternate keys are those candidate keys which are not the Primary key.
- 🔑 Foreign key is an attribute which is a Primary key in its parent table, but is included as an attribute in another host table.

	<u>empno</u> <sup>SK</sup>	<u>ename</u> <sup>SK</sup>	<u>addr</u> <sup>SK</sup>	<u>job</u> <sup>SK</sup>	<u>dept</u> <sup>SK</sup>	<u>aadhar</u> <sup>SK</sup>	<u>passport</u> <sup>SK</sup>	<u>email</u> <sup>SK</sup>	<u>phone</u> <sup>SK</sup>	
PK →	1	A	~~~~~	analyst	10	1234	11	a@x.com	~	
↙ CK	2	B	~~~~~	analyst	10	2345	22	b@x.com	~	
	3	C	~~~~~	mgr	20	3456	33	c@x.com	~	
	4	D	~~~~~	mgr	20	4567	44	d@x.com	~	
	5	E	~~~~~	clerk	30	5678	55	e@x.com	~	

key  
↓  
super key  
↓  
(unique) Candidate key  
↓  
primary key



# De-normalization

- Normalization will yield a structure that is non-redundant.
- Having too many inter-related tables will lead to complex and inefficient queries. ✓
- To ensure better performance of analytical queries, few rules of normalization can be compromised. → eg. adding computed columns, ...
- This process is de-normalization.



# Codd's rules

- Proposed by Edgar F. Codd – pioneer of the RDBMS – in 1980.
- If any DBMS follow these rules, it can be considered as RDBMS.
- The 0<sup>th</sup> rule is the main rule known as “The foundation rule”.
  - For any system that is advertised as, or claimed to be, a relational data base management system, that system must be able to manage data bases entirely through its relational capabilities.
- The rest of rules can be considered as elaboration of this foundation rule.

maths  
↓  
Set theory



# Codd's rules

- Rule 1: The information rule:
  - All information in a relational data base is represented explicitly at the logical level and in exactly one way – by values in tables.
- Rule 2: The guaranteed access rule:
  - Each and every datum (atomic value) in a relational data base is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name.
- Rule 3: Systematic treatment of null values:
  - Null values (distinct from the empty character string or a string of blank characters and distinct from zero or any other number) are supported in fully relational DBMS for representing missing information and inapplicable information in a systematic way, independent of data type.



# Codd's rules

- Rule 4: Dynamic online catalog based on the relational model:
  - The <sup>metadata</sup> data base description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data. *es. routines, user, ...*
- Rule 5: The comprehensive data sublanguage rule: – SQL
  - A relational system may support several languages. However, there must be at least one language that supports all functionalities of a RDBMS i.e. data definition, data manipulation, integrity constraints, transaction management, authorization.



# Codd's rules

- Rule 6: The view updating rule:
  - All views that are theoretically updatable are also updatable by the system.

*→ simple views*
- Rule 7: Possible for high-level insert, update, and delete:
  - The capability of handling a base relation or a derived relation as a single operand applies not only to the retrieval of data but also to the insertion, update and deletion of data.

*bulk deal ↑*
- Rule 8: Physical data independence:
  - Application programs and terminal activities remain logically unbroken whenever any changes are made in either storage representations or access methods.
- Rule 9: Logical data independence: *→ use views*
  - Application programs & terminal activities remain logically unbroken when information-preserving changes of any kind that theoretically permit un-impairment are made to the base tables.



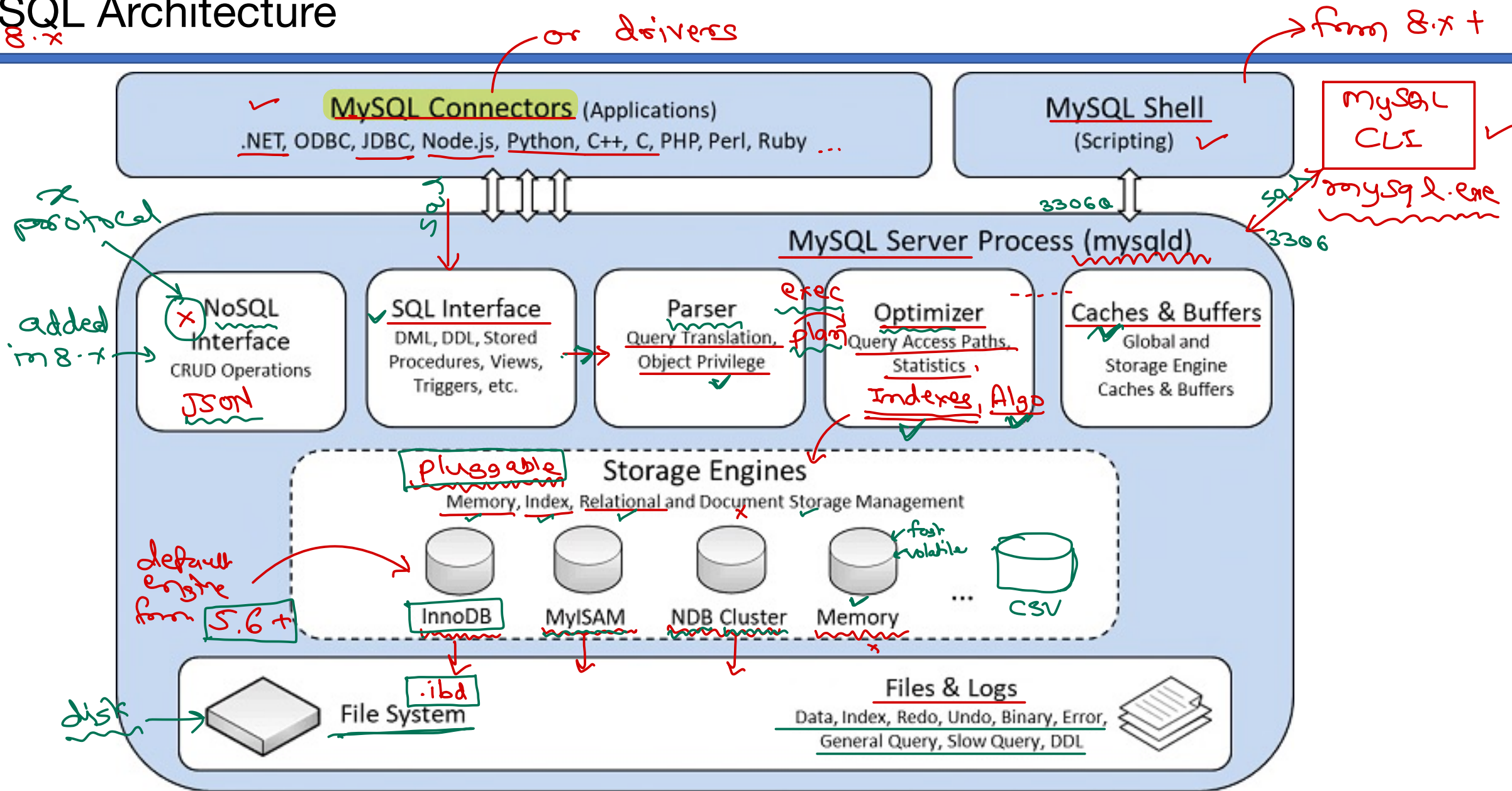
# Codd's rules

- Rule 10: Integrity independence:
  - Integrity constraints specific to a particular relational database must be definable in the relational data sublanguage and storable in the catalog, not in the application programs.
- Rule 11: Distribution independence:
  - The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only.
- Rule 12: The non-subversion rule:
  - If a relational system has a low-level (single-record-at-a-time) language, that low level cannot be used to subvert or bypass the integrity rules and constraints expressed in the higher level relational language (multiple-records-at-a-time).





# MySQL Architecture





Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>

