**--------Top SQL And Ans--------**

Q-S. What in the primary difference PL/SQLY

Ans->

PL/SQL is an advanced form of SQL developed by Oracle in sarly 90's as a superset of 501 many additional programming features to enable application development at the database level

Q-2. What is the main difference between Primary Key, Unique Key, and Foreign Key?

Ans->Following are the key differences between Primary Key, Unique Key, and Foreign Key

Primary Key:

The primary key cannot have a NULL value.

Every table can have only one primary key.

By default, Primary key supports clustered index. Thus data in the database table are physically organized

the sequence of clustered index. It can be related to another table as a Foreign Key.

It supports the generation of ID automatically with the help of Auto Increment field.

Unique Key:

Unique Constraint may have a NULL value.

Each table can have more than one Unique Constraint.

By default. Unique kev is a unique non-clustered index.

Uniqus Kev

23 Unique Cams have a NULL

22 Each table tame than one

23 By default, tinique key 24 is a uniqu

25 It is not related to another table as a Foreign Key.

26 Unique 27 Foreign Key:

Constraint doesn't support Auto Increment value.

28 A Foreign key is a field in a table whereas, it is the primary key in another table 29

30 It can accept multiple null values.

31 A Foreign key does not automatically create an index, clustered or non-clustered. You must manually cre

index on the foreign key.

32 We can have more than one foreign key in a table.

33

34 There are advantages of having a foreign key supported with a clustered indes, but you get only one per tab The advantage using a clustered index is that, on selecting the parent plus all child records, It can bring w

child records next to each other. 35 The Foreign key shouldn't have a null value. Else, the system will consider it as an uryhan recard.

37

38 39 Q-3. What is the purpose of a JOIN statement? Also, explain the different types of 3014 clauses supported in

SQL. 40 Ans->JOIN kevword is used to fetch data from two or more related tables. It returns tows wh

Q-3 at in the purpose of

SQL 40 An-2019 med is used

41

301N stat

to letos

least one mats in both the tables cut the yi

42 SQL specifies five types of 301N clauses as follows: 43

44

45 1. INNER JOIN (also 46 called as "simple join"): It returns all the rows for

which there is at least one match mentioned then "INNER JOIN works as the default join. in BOTH the tables. Il join type is not specifically

47

48 SQL Syntax for INNER JOIN:

49

50 SELECT column\_name(s)

51 FROM table1

52 INNER JOIN table2

53 ON table1.column\_name=table2.column\_name;

54 55 2. LEFT JOIN (or LEFT OUTER JOIN):

56 Returns all rows from the left table, and the matching rows from the right table. Thus the result will contain records from the left table, even if the JOIN condition doesn't find any matching records in the right table. This means, that if the "ON" clause does not match to any records in the right table, the 201 will return a rowin the result for that record in the left table, but with NULL in each column from the right table.

SQL Syntax for LEFT JOIN

SELECT coturm mm\_name(s)

162 FROM tabled

6) LEFT JOIN tablez

DA ON table1.column\_namestable2.column\_name 05

66 3. RIGHT JOIN (or RIGHT OUTER JOIN):

57 It returns all rows from the right table and the corresponding matthing rows from the left at opposite of the LEFT JOIN. Thus the result will contain all the records from the condition doesn't find any matching records in the left table. This means, that if the "ON" match to any records in the left table, the JOIN will return a row in the result for that recard in the rig but with NULL in each column from the left table.

68 69 SQL Syntax for RIGHT JOIN:

70

71 72 SELECT column\_name(s)

73 FROM table1

74 RIGHT JOIN table2

75 ON table1.column\_name=table2.column\_name;

76

77 4. FULL JOIN (or FULL OUTER JOIN): 78 It returns all the rows for which there is a match in either of the tables. Fundamentally, a FULL 20

combination of the effect produced by

both a LEFT JOIN and a RIGHT 2014. Thus we can ser att

4. FULL 30TH FULL OUTER JOIN

combinatum a

29

which there

the effect produced b

is equivalent 16 performing

Lables, Fundamentally, FULL 20

RIGHT JOIN Thus we can say that in pro

UNION left and right outer queries

151

80 SQL Syntax for FULL OUTER JOIN:

82 SELECT column\_name(s)

83 FROM table1

84 FULL OUTER JOIN table2

85 ON table1.column\_name=table2.column\_name; 86

87 5. CROSS JOIN:

88 It returns a result set which is the multiplication of the number of rows in the first and the second lable. If we do not apply WHERE clause along with CROSS JOIN, it returns the Cartesian Product. However, if

WHERE clause along with CROSS JOIN, it functions like an INNER JOIN 89 An alternative way of achieving the same result is to use column names separated by commas after SELECT mentioning the table names involved, after a FROM clause.

90

91

92 SQL Syntax for CROSS JOIN:

93

94 SELECT column\_name(s)

95 FROM table1

96 CROSS JOIN table2:

Redmi Note 10T 5G | ANI PATEL

18/22024

SQL Synthur CROSS Jom

SELECT

95 FROM table

96 CROSS JOIN table;

97 6. SELF JOIN:

98. It is used to join a table to itself as if the table were two tables. To achieve this, we temporarily rename the tables in the SQL statement.

99

100 Syntax:

101

102

103 SELECT column\_name(s)

104 FROM table1, table2

105 WHERE table1.common\_field = table2.common\_field;

106

107

108 Q-4. What is Normalization and how does it work?

109

110 Ans->It is the process of designing database tables to minimize the data redundancy is called normalita

111 112

113 We need to divide a database into two or more tables and define relationships between them.

Wenke diviteetan into two

112 Q-5. What are the various forms of Normalizationt 11日

119 Ans->Database normalization process provides following forms: 120

121 1. First normal form (1NF):

122 As per the rule of 1NF, an attribute(column) of a table can not hold multiple values. It should contam sy atomic values.

123 For Example suppose a company stores details of its employees including the name, address, and the ontact number. It is possible that some employees have more than one contacts. In that case, employee talle this:

124

125 emp\_id emp\_name emp\_address emp\_contact

126 001 Nidhi Gurgaon 9873456789,

127 9990022334

128 002 Prakash New Delhi 7838777343

129 003 Mallika New Delhi 7838005674,

130 8876453212

131 Since multiple contact numbers are for the same employees so it gets stored in the same field.

132 As per the 1NF rule, the above table is not in 1NF. To make the table as 1NF compliant, we have to stare unl single contact in one row as:

single contact

134 135 empl

emp name emp address emp\_contact

136 001 Nidhi Gurgaon 9873456789

137 001 Nidhi Gurgaon 9990022334

138 002 Prakash New Delhi 7838777343 139 003 Mallika New Delhi 7838005674

141

140 003 Mallika New Delhi 8876453212

142 2. Second normal form (2NF):

143 A table is in 2NF if the following conditions hold true:

144

145 The table is in 1NF.

145 No non-prime attribute is dependent on the proper subset of any candidate key of the table 147

148 An attribute that is not part of any candidate key is known as a non-prime attribute. 149

150 For Example, a school stores data about the teachers and the subject they teach. A teacher can teach t one subject. Thus the table will look like:

151 52 teacher\_id subject teacher\_age

53 01A Maths 40

54 01A Physics 40

teacher 10

subject teacher Lage

01A Mats 40 154 014 Physics 40

155 0XB English 42

154 01C Chemistry 40 457 OIC EVS 40

158 Candidate Keys:

155 (teacher\_id, subject)

160

161

167 Nonprime attribute:

163 teacher age

164

165 The table is in 1 NF. However, it is not in 2NF because nonprime attribute teacher age is dependenton teacher\_id alone which is a proper subset of the candidate key.

166

167 Now to make the table as 2NF compliant, we break the table as follows:

168

169

170 Teacher\_details table:

171 teacher\_id teacher\_age

172 01A 40

173 018 42

174 010 40

125 Teacher subiect table:

Teacher

172 01A 40

173 016 42

174 010 40

175 Teacher subject table:

176 teacher id subject

177 01A Maths

178 01A Physics 179 018 English

180 01C Chemistry

181 01C EVS

182

183 3. Third normal form (3NF):

184 A table is in 3NF if both the given conditions hold true:

185

186 The table is in 2NF.

187 For every functional dependency (X-> Y), at least one of the following conditions huld

188 X is a super key of the table.

189 Y is a prime attribute of the table.

190 A prime attribute is an attribute that is part of the candidate key. 191 Suppose we have a table that stores information about employee address as:

192

193 emp\_id emp\_name emp\_city emp\_state emp\_zip

001 Mavank Ghaziabad UP 201001

002 Skstant Dnalier HP 22299

196 003 Malik 197 Maryana 122001 In the above table following are the

196

199 Super keys:

200 (emp\_id), (emp\_id, emp\_name), (emp\_id, emp\_name, emp\_zip)...som 201

202 Candidate Keys:

203 (emp\_id} 204

205 Non-prime attributes:

206 all attributes except emp\_id are non-prime as they are not part of any candidate keys. I

207

208 Here the non-prime attributes emp\_state, emp\_city is dependent on emp zip which is dependent un mo Thus this created a transitive dependency on the super key emp id which is a violation of the 310 rule. To

make it 3NF compliant, break the table as follows:

209

210 emp\_id emp\_name emp\_zip

211 001 Mayank 201001

212 002 Saksham 222999

213 003 Mallika 122001

214 and

215

216 emp id emo citv emo state emo zin

002 Gwalior HP 222999 219 003 Gurgaon Haryana 1220001

220

221 4. Boyce-Codd normal form ( BCNF):

222 It is an advanced version of 3NF also called as 3.5NF. BCNF is stricter than 3N7. 223

224 A table is BCNF compliant if it is in 3NF and for every functional dependency X->Y, I should be the superbay the table.

225 226

227

228 Q-6. What is the difference between a superkey and the candidate key?

229

230 Ans->A superkey is a combination of columns that uniquely identifies any row within a relational database management system (RDBMS) table.

231

232 Whereas, a candidate key is a superkey containing a minimum number of columns that can uniquelydenty each row.

233

234

235

236 0-7. What are the different types of statements available in SOL?

330 Anz: DML (Data Manipulation Language

240 These are used to manage records in the tabile. It includes the basic operations carried out un the tabutar dale like selecting few records, inserting new ones, deleting the unnecessary one existing ones. Following are different DML statements available in SQL ones, and updating) modifying the

241

242 <SELECT>- retrieve data from the database

243 <INSERT> - to insert data into a table

244 <UPDATE> - it updates existing data within a table 245 <DELETE> - to delete all records from a table

246 <MERGE> - UPSERT operation (insert or update)

247 <CALL> - to call a PL/SQL or Java subprogram

248 <EXPLAIN PLAN> - define access path to data

249 <LOCK TABLE> control concurrency

250

251 2. DDL (Data Definition Language):

252 DDL statements are used to alter/modify a database or table structure and schema. These statements hand the design and storage of database objects. Following are different DDL statements available in SQL

253

254 <CREATE> - to create objects in the database

255 <ALTER> - alters the structure of the database

256 <DROP> to delete objects from the database

257 <TRUNCATE - remove all records from a table. It also frees all the space allocated to them

<TRUNCATE> - remove all records from a table. It also frees all the space allocated to them 258 <COMMENT>add comments to the data dictionary

259 <RENAME> - to rename an object.

260

261 3. DCL (Data Control Language):

262 DCL statements control the level of access that users have to the database objects. Following are different DCL statements available in SQL:

263

264 <GRANT> - it gives access privileges to the user for the database

265 <REVOKE> - to withdraw the access privileges given by GRANT command.

266

267 4. TCL (Transaction Control Language):

268 It allows you to control and manage transactions to maintain the integrity of data within SQL statements Following are different TCL statements:

269 270 <COMMIT> to save the work

271 <SAVEPOINT> - identify a point in a transaction to which you can rollback at a later point in time when

271 <SAVEPOINT>- identify a point in a transaction to which you can rollback at a later pain in Time when required

272 <ROLLBACK> - restore the database to original since the last COMMIT 273 <SET TRANSACTION> - Change transaction options like isolation level and what rollback segment to un

274

275

276 Q-8. What is the role of COMMIT in an SQL transaction?

277

278 Ans->COMMIT finalizes the changes, introduced by all SQL statements included in the transaction as permanent in the database.

279

280 Thus the changes made by the SQL statements of a transaction become visible to other user's session transactions that start only after the transaction gets committed.

281

282

283

284 Q-9. What are the common properties of a database transaction?

285

286 Ans->Following are the properties on which every database depends to perform reliable transactions. Ws

Atts-Tollowing are the properties.o usually call them as ACID properties whey cause depends to perform reliable transactio

267

289

288 Atomicity: A transaction may contain two or moce discrete pieces of information. Mamicity means wither commit all the data or nothing.

290 Consistency: A transaction creates a new and valid state of data. However, if any Tallure scurs, if reverts the data to its original state before the start of the transaction. 291

292 Isolation: A transaction under execution and not yet committed must remain isolated from any other transaction.

293

294 Durability: System stores the committed data so that the data is available in its currect state, in case табл or system restart happens.

295

296

297 Q-10. What are the main points that differentiate between the "delete", "truncate" and "drop" commands

298 Ans.

299

300 A. DELETE:

301 It is a DML statement.

302 It applies a filter based on an optional WHERE clause to identify the rows that will get deleted

317 DELETE FRON salite me

3156 TRUNCATE 319 It is a DOL Statement

320 Removes all rows from a table and it becomes empty, But, the table structures, its columns, constraints, вид

indexes remain intact.

321 It is not possible to roll back the TRUNCATE transaction.

322 It resets the identity of table i.e. the auto-incrementing keys are reset to 1. It's just like having a brand new table.

323 It is faster than DELETE and uses a lesser amount of system and transaction logs. 324 TRUNCATE cannot be used on a table referenced by a FOREIGN KEY constraint.

325 No Triggers will get fired. 326 Cannot use WHERE conditions.

327 Use this when you just want an empty table.

328 SYNTAX:

329 TRUNCATE TABLE table\_name

330 C. DROP:

331 It is a DDL statement.

332 It not only removes the table from the database. Its structures, indexes, privileges, and constraints a

removed. 333 It is not possible to roll back the DROP transaction.

334 No Triggers will get fired.

335 Use this when you don't need that table anymore.

137 DROP TABLE

340 Q-11. What is an Index? Explain the different types of index 341 Ans.

342

143 An index is a performance enhancement method that allows faster retrieval of records from the table. An index 344 creates an entry for each value thus making data retrieval faster.

346

345 While creating an index, we should remember the columns which will be used to make SQL queries and create one or more indexes on those columns.

347 Following are the available indexes.

348

349 1. Clustered index:

350 It sorts and stores the rows of data in the table or view, based on its keys. These are the columns included the index definition. There can be only one clustered index per table because sorting of data rows can be only in one order.

351 352 2. Nonclustered index:

353 It contains the nonclustered index key value and each key value entry, in turn, has a pointer to the data re Thus a nonclustered index contains a pointer to the physical location of the record. Each talile can have 199 nonclustered indexes.

354

354

It contains nomustard index kry

Thus Heuricatered

Index contains

nonciustered indexes,

355 3. Unique Indexi

value entry,

physical location

in turn, has a poinder

to the Esta fon

of the record. Each table can have 399

356 This indexing does not allow the field. Ito have duplicate values if the column is applied automatically when a primary key is defined. unique indesed.n

357

358

359

360 Q-12. What is the purpose of a Subquery?

361 Ans.

362

1

363 A Subquery also called as Nested query is a query within another SQL query a and embedded clause. A Subquery within the WHERE is always executed first and passes its result to the main query. This data acts as a ter condition in the main query to further restrict the data to be retrieved. Subqueries work with the SELECT, INSERT, UPDATE, and DELETE statements along with the operators like, <, >, >, <=, IN, and BETWEEN

364

365 Following are some important properties of a Subquery that we must know:

366

367 Always write a Subquery within a parenthesis.

368 It can contain more than one column in SELECT clause only if the main query has multiple columns. 369 We cannot use ORDER BY in a Subquery. Instead, use GROUP BY which performs the same function as ORDER

BY.

22. Nonc

953

Follow

Ant properti

AKA

Al Bulaguery within a p It can contaliere than one colum

369 We cannot use ORDER BY. in SELECT only if the main query has multiple columes BY in a Subquery. Instead, use GROUP BY which performs the same function as ORDER

370 We cannot use BETWEEN operator with a subquery but, can use it within a Subquery.

372

371 You can nest Subqueries up to 32 levels.

323

374 Q-13. What are Constraints? Explain the different Constraints available in SQL?

375 Ans.

076

377 These are the set of rules that determine or restrict the type of data that can go into a table, to maintain the accuracy and integrity of the data inside the table.

378

379 Following are the most frequent used constraints, applicable to a table:

380

381 <NOT NULL> It restricts a column from holding a NULL value. It does not work on a table 382 <UNIQUE> It ensures that a field or column will only have unique values. It is applicable to both colum and table.

383 <PRIMARY KEY> uniquely identifies each record in a database table and it cannot contain NULL values. 384 <FOREIGN KEY> It is used to relate two tables. The FOREIGN KEY constraint is also used to restrict actions

that would destroy links between tables. 385 <CHECK CONSTRAINT> It is used to restrict the value of a column between a range. Il performs a check on the

values, bef

306 <DEFAULT

387

386

Uncing them

used to rest

into the

tion

Wined to insert a default valueolumn.

389 Q-14. What is the difference between Union and Union ALL?

390 Ans.

391

392 UNION and UNION ALL merges the contents of two structurally-compatible tables into a single contined tule

393 394 The difference between UNION andUNION ALL is that UNION will remove duplicate records whereas USION ALL will include duplicate records.

395 The performance of UNION ALL is better than UNION as UNION requires the server to do additional work removing duplicates.

396 For performance reasons, it is recommended to use UNION ALL is the scenarios when it is certain that there will be no duplicates or cases where having duplicates is not a problem.

397 398

399

400 Q-15. What is a stored procedure? Discuss its advantages and disadvantages?

401 Ans.

402 403 A stored procedure is a group of SQL statements that has been created and stored in the database. Suppor

there is a query that we execute very frequently.

In that case, instead of

writing that query, again and ag

We can save it as a stored procedure and then just call the stored procedure to execute the 501,code

+

between a range. It perfecms a check

checking before saving data ints su

DQ-15. Wha

101 Ans 402

advantages

403 A stores procedure is a group of SQL that has been created and stored in the datatine Suppose there is a query that we execute very frequently. In that case, instead of writing that query, again and again We can save it as a stored procedure and then just call the stored procedure to execute the SQL code. It als allows passing parameters to the stored procedure. 404

405 An Example of Stored Procedure.

406 USE testdb;

407 GO

108 CREATE PROCEDURE test\_procedure

409

AS

410 SELECT FirstName, LastName FROM testdb;

411

GO EXEC test\_procedure;

412 413

414

GO

415

DROP PROCEDURE test\_procedure;

GO

416

417 Following are the advantages of using stored procedure:

418

419 A stored procedure allows modular programming. It means once we create a stored procedure and sture it in the database, then we can call it any number of times as per our requirement. 420 It allows faster execution in the case when the operation executes same. SOL code repetitive. It

000

Redmi Note 10T 5G | ANI PATEL

18/03/2024

420 It allows faster execution in the case when the and optimized during its first execution. uperation executes same SQL code repetitively gets surand A compiled version of the stored procedure remains in mem for later use resulting in much faster execution time.

421 Stored Procedure can reduce network traffic. Suppose there is code, creates a an operation tha that requires executing large SQL stored procedure. Thus rather than sending hundreds of lines of code over the natwurs

to send a single statement that executes the code. 122 Stored procedures provide better security to your data. A user might be allowed to trigger a stare p even if he doesn't have the permission to execute the procedure's statements directly. and thus utilizes more memory in the daава

423 Its disadvantage is that the execution requires a

server.

424

425

426

427 Q-16. What is a View? What are its advantages and disadvantages?

428 Ans.

429

430 A View is a virtual table which contains data from one or more tables. It selects only required values thus restricting the access to table data. And it also makes complex queries a bit easier.

431

432 Following are the advantages of using Views:

433 It enables viewing data without storing the data in an object. 434 Restrict the view of a table by hiding some of its columns.

435 Join two or more tables and display it as a single object.

436 Restrict the access of a table so that nobody can insert rows in the table without permission.

137 Disadvantages of Views:

438 Cannot apply DML statements on it.

439 A View becomes inactive if a table that is a part of View gets dropped.

440 It is an object and hence, it consumes memory.

441

442 443 Q-17. List the built-in functions, available in SQL?

444 Ans.

445

446 Following are the important built-in functions available in SQL.

447

448 AVG(): Returns the average value. of rows.

449 COUNT(): Returns the number 450 FIRST(): Returns the first value.

449 COUNT(): Returns the number of rows. 450 FIRST(): Returns the first value.

451 LAST(): Returns the last value.

452 MAX(): It gives the largest value as output.

153 MIN(): It gives the smallest value as output.

454 SUM(): Outputs the Sum.

455 UCASE(): Converts a value to upper case.

456 LCASE(): Converts a value to lower case.

457 MID(): Extract the middle character from a string or number. 458 LEN(): Returns the length of a text field.

459 ROUND(): Round Off a numeric field to the number of decimals specified.

460 NOW(): Returns the current system date and time.

461 FORMAT(): defines how a field is to be displayed.

462

463

464 Q-18. What are Triggers? What are its benefits? Can we invoke a trigger explicitly?

465 Ans.

467 The trigger is a type of stored program, which gets fired automatically when some event occurs, We writs a Trigger as a response to either of the following event:

468

469 A database manipulation (DML) statement (DELETE, INSERT, or UPDATE)

470 A database definition (DDL) statement (CREATE, ALTER, or DROP). 471 A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN)

472 SQL allows defining Trigger on the table, view, schema, or database associated with the event.

473 Following are its benefits:

475 Generating some derived column values automatically.

476 Enforcing referential integrity.

477 Event logging and storing information on table access.

478 Auditing.

479 Synchronous replication of tables.

480 Imposing security authorizations.

481 Preventing invalid transactions.

482 It is not possible to invoke a trigger explicitly. It gets invoked automatically if an event gets execsted on the table having an association with the trigger.

483

Q-110

487

Ans

the Pleeves

available

48%

489 Following ans some of the frequently used Clauses in SQL:

490

491 1. WHERE:

493

492 Using WHERE clause, we can specify selection criteria to select required records from a table

494 Syntax:

495

496 SELECT field1, field2,...fieldN table\_name1, table name2...

497 [WHERE condition1 [AND [OR]] condition2......;

498 2. ORDER BY:

499 It is used to sort the records in your result set.

500

501 Syntax:

502

503 SELECT expressions

504 FROM tables

505 [WHERE conditions]

506 ORDER BY expression [ASC | DESC];

507 3. TOP:

508 It specifies the number of records to return. This clause is valuable for large tables with thousands of records

508 It specifies the number of records to return. This clause is valuable for large tables with thousands of rend 509

510 Syntax: 511

512 SELECT TOP number percent column\_name(s)

513 FROM table\_name;

514 4. GROUP BY:

515 It is used to group values from a column, and, if required, perform calculations on that column. It applies t aggregate functions such as SUM, AVG, MAX, MIN, and COUNT.

516

517 Syntax: 518

519 SELECT

520 c1, c2,..., cn, aggregate\_function(ci)

521 FROM

522 table

523 WHERE

524 where\_conditions 525 GROUP BY c1, c2,...,cn;

526 5. HAVING:

5221 H SELECT statement το χρns for a group of rows or aggregates

529 The MySQL HAVING clause is Frequently used with the GROUP BY clause to apply a filter condition to The columns that appear in GROUP BY clause. The HAVING clause behaves like the WHERE clause in cast the GROUP BY clause gets excluded.

530

531 Syntax:

532

533 SELECT column\_name, aggregate\_function(column\_name)

534 FROM table\_name

535 WHERE column\_name operator value

536 GROUP BY column\_name

537 HAVING aggregate\_function(column\_name) operator value;

538 6. UNION:

539 It is used to combine the result-set of two or more SELECT statements. It is important to take care that ac SELECT statement within the UNION has the same number of columns. Also, these columns have same data types or convertible data type and their order in the SELECT statement is also same.

540

541 Syntax:

542

543 SELECT column\_name(s) FROM table1

544 UNION [DISTINCT | ALL]

545 SELECT column name(s) FROM table2:

556 Q-20. What is the purpose of isolation levels in SQL?

557 Ans.

558

559 Transactions use an isolation level that specifies the extent to which a transaction must be isolated from any data modifications caused by other transactions. These also help in identifying which concurrency side-s are permissible.

560

561 Please refer the below list for more clarity on the different type of levels.

561 Please refer the below list for more clarity on the different type of levels.

562

563 i-> Read Committed.

564 It ensures that SELECT query will use committed values of the table only. If there is any active transaction on the table in some other session, then the SELECT query will wait for any such transactions to complete Read Committed is the default transaction isolation level.

565

566 ii-> Read Uncommitted.

567 There is a transaction to update a table. But, it is not able to reach to any of these states like complete, comm or rollback. Then these values get displayed (as Dirty Read) in SELECT query of "Read Uncommitted solatio transaction.

568

569 III-> Repeatable Read.

1

570 This level doesn't guarantee that reads are repeatable. But it does ensure that data won't change for the life of the transaction once.

571

572 iv-> Serializable.

573 It is similar to Repeatable Read level. The only difference is that it stoos Phantom Read and utilizes the ran

573 It is similar to Repeatable Read level. The only difference is that it stops Phantom Read and utilizes the range lock. If the table has an index, then it secures the records based on the range defined in the WHERE (like where ID between 1 and 3). If a table does not have an index, then it locks complete table Gene

574

I 575 v-> Snapshot.

576 It is similar to Serializable isolation. The difference is that Snapshot does not hold a lock on a table during the transaction. Thus allowing the table to get modified in other sessions. Snapshot isolation maintains version in Tempdb for old data. In case any data modification happens in other sessions then existing transaction displays the old data from Tempdb.