1. Różnice między klauzulami WHERE i HAVING:

· WHERE: Używana do filtrowania danych na poziomie poszczególnych wierszy w tabeli na podstawie określonych warunków.

· HAVING: Używana do filtrowania danych na poziomie grup wierszy, które powstały w wyniku użycia klauzuli GROUP BY.

2. Funkcje agregujące i funkcje okna:

· Funkcje agregujące (np. `SUM`, `AVG`, `MIN`, `MAX`) są stosowane do obliczania wartości na podstawie zestawu wierszy w tabeli.

· Funkcje okna wykonują operacje agregacyjne na zestawie wierszy w obrębie okna, umożliwiając obliczenia w kontekście okna zamiast całego zestawu wynikowego.

3. Łączenie wyników z różnych zapytań:

· UNION: Łączy unikalne wiersze z różnych zapytań i usuwa duplikaty.

· UNION ALL: Łączy wiersze z różnych zapytań i zachowuje duplikaty.

· INTERSECT: Zwraca wspólne wiersze z różnych zapytań.

4. Różnice między LEFT JOIN a JOIN:

· JOIN (INNER JOIN): W przypadku braku dopasowania, wiersze z jednej z tabel są pomijane.

· LEFT JOIN (LEFT OUTER JOIN): Wiersze z lewej tabeli są zawsze zwracane, a brakujące wartości z prawej tabeli są uzupełniane NULLami.

5. Klauzula ORDER BY vs. GROUP BY:

· ORDER BY: Sortuje dane w kolejności rosnącej lub malejącej.

· GROUP BY: Grupuje wiersze na podstawie wartości kolumn, często w połączeniu z funkcjami agregującymi.

6. Indeksowanie:

· Indeksowanie może poprawić wydajność zapytań SQL poprzez zmniejszenie liczby wierszy skanowanych podczas wykonywania zapytania. Pomaga to w szybszym wyszukiwaniu danych.

7. Indeks unikalny vs. indeks zwykły:

· Indeks unikalny: Zapewnia, że wartości w kolumnie są unikalne.

· Indeks zwykły: Nie wymaga unikalności wartości w kolumnie.

8. Tworzenie kopii zapasowej bazy danych w SQL Server Management Studio:

· Otwórz SQL Server Management Studio, przejdź do sekcji Bazy danych, kliknij prawym przyciskiem na bazę danych, wybierz Zadania > Utwórz kopię zapasową.

9. Aktualizacja danych w SQL:

· Modyfikację istniejących danych wykonuje się za pomocą polecenia `UPDATE`.

10. Wybór bazy danych:

· MySQL i PostgreSQL: Odpowiednie dla dużych aplikacji z zaawansowaną funkcjonalnością.

· SQLite: Idealne dla prostych aplikacji, lokalnych baz danych i środowisk z ograniczonymi zasobami.

11. Usuwanie duplikatów:

· Aby usunąć duplikaty, można posortować dane przy użyciu `ORDER BY`, aby wybrać rekordy do usunięcia lub zachowania.

12. Brakujące dane w SQL:

· Brakujące dane są reprezentowane jako `NULL`. Do ich sprawdzania używa się `IS NULL` i `IS NOT NULL`, ponieważ `NULL` nie jest wartością i nie można używać operatorów porównawczych.

13. ISNULL vs. COALESCE:

· ISNULL: Zastępuje `NULL` jedną określoną wartością.

· COALESCE: Zastępuje `NULL` pierwszą nie·nullową wartością z listy.

14. Dodawanie kolumn do tabeli:

· Użyj `ALTER TABLE ... ADD COLUMN`, aby dodać nowe kolumny do istniejącej tabeli.

15. Tworzenie tabeli:

· Użyj `CREATE TABLE`, aby utworzyć nową tabelę z określoną nazwą i atrybutami kolumn.

16. Transakcje w SQL Server:

· Tworzenie transakcji za pomocą `BEGIN TRANSACTION`, `COMMIT TRANSACTION`, i `ROLLBACK TRANSACTION`. Transakcje rozproszone są wymagane, gdy obejmują różne menedżery zasobów.

17. Klucz główny vs. klucz obcy:

· Klucz główny: Unikalnie identyfikuje wiersz w tabeli.

· Klucz obcy: Odnosi się do klucza głównego w innej tabeli.

18. Monitorowanie i audyt indeksów:

· Używaj narzędzi do monitorowania indeksów i regularnie przeglądaj schemat bazy danych, aby optymalizować wydajność.

19. Typy danych CHAR vs. VARCHAR:

· CHAR: Używany, gdy długość danych jest stała.

· VARCHAR: Używany, gdy długość danych jest zmienna.

20. Funkcje agregujące:

· `COUNT`: Zwraca liczbę rekordów.

· `AVG`: Zwraca średnią wartość.

· `SUM`: Zwraca sumę wartości.

· `MAX`: Zwraca maksymalną wartość.

21. Ograniczanie wyników:

· LIMIT: Ogranicza liczbę wierszy w wynikach po sortowaniu (MySQL, PostgreSQL, SQLite).

· ROWNUM: Ogranicza liczbę wyników przed sortowaniem.

· FETCH FIRST: Ogranicza liczbę wyników po sortowaniu (Oracle).

22. Modyfikowanie kolumn:

· `ALTER TABLE` służy do dodawania, usuwania lub zmiany kolumn w istniejącej tabeli.

23. Poziomy normalizacji:

· 1NF: Usuwa wielowartościowe komórki.

· 2NF: Usuwa częściowe zależności od klucza głównego.

· 3NF: Usuwa przechodnie zależności między kolumnami.

24. Sprzężenie zewnętrzne:

· (+): Operator sprzężenia zewnętrznego w SQL Server.

25. Funkcja AVG:

· AVG(nazwa\_kolumny): Oblicza średnią wartość kolumny w tabeli.

26. Ograniczanie wyników w różnych systemach:

· TOP: SQL Server, Sybase.

· LIMIT: MySQL, PostgreSQL, SQLite.

27. Sortowanie danych w SQL Oracle:

· Aby sortować dane malejąco, użyj klauzuli `DESC` w `ORDER BY`.

28. Formatowanie i konwersja dat:

· STR\_TO\_DATE: Zamienia ciąg znaków na datę według formatu.

· DATE\_FORMAT: Zmienia format daty.

· TO\_DATE: Zamienia ciąg znaków na datę.

· CONVERT: Zmienia format tekstowy na datę.

29. Tworzenie indeksu:

· Użyj `CREATE INDEX` do definiowania indeksu dla tabeli.

30. Aktualizacja wielu kolumn:

· `UPDATE tabela SET kolumna1 = wartość1, kolumna2 = wartość2 WHERE warunek;`

31. Funkcje vs. procedury:

· Funkcje: Zwracają wartości używane w dalszych obliczeniach.

· Procedury: Nie zwracają wartości, lecz wykonują operacje.

32. Blokady w SQL Server:

· Użyj widoku `sys.dm\_tran\_locks` do uzyskania informacji o blokadach.

33. Operator IN vs. EXISTS:

· IN: Dopasowuje wartości do listy wartości.

· EXISTS: Sprawdza istnienie wierszy w podzapytaniu.

34. Operatory arytmetyczne:

· `+`, `·`, `\*`, `/` służą do wykonywania obliczeń w zapytaniach SQL.

35. Transpozycja w SQL:

· Zamienia wiersze na kolumny i odwrotnie, używając `SELECT`, `CASE` i `GROUP BY`.

36. Funkcje DateAdd i DateDiff:

· DateAdd: Dodaje jednostki do daty/godziny.

·

DateDiff: Zwraca różnicę między dwiema datami/godzinami.

37. Usuwanie kolumn:

· `ALTER TABLE ... DROP COLUMN` usuwa kolumnę z tabeli.

38. LIKE vs. IN:

· LIKE: Używa symboli wieloznacznych do wyszukiwania podobnych wartości.

· IN: Umożliwia dopasowanie wartości do listy określonych wartości.

39. Tworzenie indeksu:

· `CREATE INDEX` definiuje nazwę indeksu, tabelę i kolumnę/kolumny do indeksowania.

40. Używanie indeksów:

· Indeksy mogą poprawić wydajność zapytań przez szybszy dostęp do danych. Używaj ich dla kolumn często używanych w klauzulach WHERE, JOIN i ORDER BY.