

## 1. View vs Stored Procedure الفرق بين:

:View

تعريف: هو استعلام SQL محفوظ يتم التعامل معه كجدول افتراضي. يمكن استخدامه لتجميع البيانات من جداول متعددة وتبسيط الاستعلامات المعقدة.

الفائدة: تبسيط قراءة البيانات بدون الحاجة لكتابة استعلامات معقدة في كل مرة.

:Stored Procedure

تعريف: هو مجموعة من أوامر SQL مخزنة يمكن تنفيذها مباشرة في قاعدة البيانات.

الفائدة: تحسين الأداء من خلال تنفيذ عدة عمليات في قاعدة البيانات دفعة واحدة، كما يمكن استخدامه لتنفيذ مهام معقدة مثل التحديثات والحذف.

## 2. Trigger ما هو:

تعريف: Trigger هو نوع من الإجراءات المخزنة يتم تنفيذه تلقائيًا استجابةً لأحداث معينة على جدول معين، مثل عمليات الإدراج أو التحديث أو الحذف.

الاستخدام: يتم استخدامه لضمان نزاهة البيانات، مثل تحديث الحقول التلقائية أو إنشاء سجلات مرتبطة في جداول أخرى.

## 3. Where vs Having الفرق بين:

:Where

تعريف: يتم استخدامه لتصفية الصفوف قبل تجميع البيانات في استعلام SQL.

الاستخدام: يستخدم عادةً مع استعلامات `SELECT` لتحديد الصفوف التي سيتم تضمينها في التجميع.

:Having

تعريف: يتم استخدامه لتصفية مجموعات البيانات بعد عملية التجميع.

الاستخدام: يستخدم مع العبارات التجميعية (`GROUP BY`) لتصفية المجموعات بعد تجميع البيانات.

## 4. Functions أنواع:

Scalar Functions: ترجع قيمة واحدة (مثل `GETDATE`).

Table-Valued Functions: ترجع مجموعة بيانات (جدول) ويمكن استخدامها مثل الجداول العادية في الاستعلامات.



Aggregate Functions: تقوم بأداء عمليات حسابية على مجموعة من القيم وترجع قيمة واحدة (مثل

## 5. Constrain ما هو:

تعريف: القيود (Constraints) هي قواعد تُطبق على الأعمدة في قاعدة البيانات لتقييد أنواع البيانات التي يمكن إدخالها في العمود.

أنواع القيود:

NOT NULL: يضمن أن العمود لا يمكن أن يحتوي على قيم فارغة.

UNIQUE: يضمن أن كل القيم في العمود فريدة.

PRIMARY KEY: يضمن أن كل صف فريد، ويستخدم لتعريف الأعمدة الأساسية في الجدول.

FOREIGN KEY: يضمن أن القيم في العمود تتطابق مع القيم في عمود آخر (غالبًا من جدول آخر).

## 6. Code First vs DB First الفرق بين:

Code First:

تعريف: يتم تصميم النموذج (Model) في الكود أولاً، ومن ثم يتم توليد قاعدة البيانات من الكود. الفائدة: يوفر مرونة للمطورين الذين يفضلون العمل مع الكود.

DB First:

تعريف: يتم تصميم قاعدة البيانات أولاً، ومن ثم يتم توليد النموذج (Model) من قاعدة البيانات. الفائدة: يناسب الحالات التي تكون فيها قاعدة البيانات موجودة بالفعل.

## 7. Union vs Union All الفرق بين:

Union: يقوم بدمج نتائج استعلامين أو أكثر، لكنه يزيل القيم المكررة.

Union All: يقوم بدمج نتائج استعلامين أو أكثر بدون إزالة القيم المكررة، مما يجعله أسرع من `Union`.

## 8. Joins أنواع:

INNER JOIN: يرجع فقط الصفوف التي لها تطابق في كلا الجدولين.

LEFT JOIN (or LEFT OUTER JOIN): يرجع كل الصفوف من الجدول الأيسر (الأول) بالإضافة إلى الصفوف المتطابقة من الجدول الأيمن.

RIGHT JOIN (or RIGHT OUTER JOIN): يرجع كل الصفوف من الجدول الأيمن (الثاني) بالإضافة إلى الصفوف المتطابقة من الجدول الأيسر.

FULL JOIN (or FULL OUTER JOIN): يرجع جميع الصفوف عندما يوجد تطابق في أحد الجدولين أو كليهما.

## 9. Middleware ما هو:

تعريف: الـ Middleware هو مكون في ASP.NET Core يتعامل مع طلبات HTTP قبل أن تصل إلى Controller وبعد أن تخرج منه، الاستخدام: يستخدم لإجراء عمليات مثل المصادقة، تسجيل الطلبات، التوجيه، ومعالجة الأخطاء.

## 10. Serialization ما هو:

تعريف: هي عملية تحويل كائن إلى صيغة يمكن تخزينها أو نقلها (مثل JSON أو XML). الاستخدام: يتم استخدام Serialization لتحويل الكائنات إلى تنسيقات يمكن إرسالها عبر الشبكة أو حفظها في ملف، ومن ثم إعادتها إلى كائنات لاحقًا باستخدام Deserialization.

## 11. Rest API ما هو:

تعريف: REST (Representational State Transfer) هو نمط معماري يستخدم لتصميم واجهات برمجية (APIs) تعمل عبر بروتوكول HTTP. الفائدة: REST APIs بسيطة ومرنة وتستخدم بشكل واسع لتبادل البيانات بين الأنظمة المختلفة.

## 12. Seed ما هو:

تعريف: Seeding في قواعد البيانات هو عملية ملء قاعدة البيانات ببيانات افتراضية أو بيانات أولية. الاستخدام: يستخدم لتوفير بيانات أولية ضرورية لتشغيل التطبيق، مثل بيانات المستخدمين الافتراضيين أو الإعدادات الافتراضية.

## 13. Normalization ما هو:

تعريف: Normalization هو عملية تنظيم قاعدة البيانات لتقليل التكرار والتأكد من أن البيانات مترابطة بشكل جيد. الفائدة: يزيد من كفاءة قواعد البيانات عن طريق الحد من التكرار وتحسين سلامة البيانات.

## 14. Composite Key ما هو:

تعريف: المفتاح المركب (Composite Key) هو مفتاح يتألف من عمودين أو أكثر ويستخدم لتحديد كل صف في الجدول بشكل فريد. الاستخدام: يستخدم عندما لا يكون عمود واحد كافيًا لتحديد الصف بشكل فريد.

## 15. Unique Key ما هو:

تعريف: Unique Key هو قيد يضمن أن كل قيمة في العمود فريدة من نوعها، لكنه يختلف عن المفتاح



تعريف: هو نمط تصميم يهدف إلى جعل الكود أكثر مرونة وقابلية للصيانة عن طريق فصل كيفية إنشاء التبعيات من الكود الذي يستخدمها.

الفائدة: يسهل اختبار الكود ويقلل من الترابط الوثيق بين الكائنات.

## 17. Unit of Work ما هو:

تعريف: هو نمط تصميم يستخدم لإدارة مجموعة من العمليات التي يجب أن تتم كمعاملة واحدة (Transaction).

الاستخدام: يستخدم لضمان أن جميع العمليات داخل معاملة يتم تنفيذها بنجاح أو يتم التراجع عنها بالكامل.

## 18. Token ما هو:

تعريف: Token هو قطعة من البيانات تستخدم في أنظمة المصادقة لتحديد المستخدمين والتحقق من هويتهم.

الاستخدام: يستخدم بشكل واسع في أنظمة المصادقة المستندة إلى الويب مثل OAuth وJWT.

## 19. Alias في SQL ما هو:

تعريف: Alias هو اسم مؤقت يمكن إعطاؤه لعمود أو جدول في استعلام SQL لتسهيل الكتابة والقراءة. الاستخدام: يستخدم لتبسيط الاستعلامات أو عندما يكون لديك جداول متعددة بنفس الأسماء.

## 20. Alias في API ما هو:

تعريف: يستخدم Alias في APIs لإعطاء مسارات أو دوال أسماء أسهل أو أكثر وضوحاً للاستخدام. الاستخدام: يسهل على المطورين التعامل مع الAPI ويسهل قراءة الكود.

## 21. DTL - Data Transfer Language ما هو:

تعريف: DTL يشير إلى اللغة المستخدمة لنقل البيانات بين مختلف النظم أو قواعد البيانات. الاستخدام: يستخدم في عمليات التكامل بين الأنظمة المختلفة أو لنقل البيانات بين قواعد البيانات.

## 22. Authorization vs Authentication الفرق بين:

Authentication: هو عملية التحقق من هوية المستخدم (من هو؟).

Authorization: هو عملية تحديد ما إذا كان المستخدم مصرحاً له بالقيام بعملية معينة أو الوصول إلى مورد معين (ماذا يمكنه أن يفعل؟).



## 23. Cluster vs Noncluster الفرق بين:

Clustered Index: يخزن البيانات الفعلية للجدول على القرص بالترتيب المحدد بواسطة الفهرس.  
Non-Clustered Index: يخزن الفهرس فقط ولا يخزن البيانات الفعلية، بل يحتوي على مؤشر يشير إلى مواقع البيانات.

## 24. Endpoint ما هو:

تعريف: هو نقطة نهاية في تطبيق الويب، عادةً ما يكون عنوان URL يمكن للعميل الوصول إليه لطلب خدمة معينة أو مورد.

الاستخدام: يستخدم لتحديد الأماكن التي يمكن للعميل إرسال الطلبات إليها في التطبيق.

## 25. Authorized Endpoint ما هو:

تعريف: هو Endpoint يتطلب مصادقة المستخدم قبل السماح له بالوصول إلى الموارد أو تنفيذ عمليات معينة.

الاستخدام: يستخدم لحماية الموارد الحساسة أو العمليات التي لا ينبغي الوصول إليها إلا من قبل مستخدمين معينين.

## :Company 2

SQL Profiling ما هو:

تعريف: SQL Profiling هو عملية تتبع وتحليل أداء استعلامات SQL التي يتم تنفيذها في قاعدة البيانات.

الاستخدام: يتم استخدامه لتحديد الاستعلامات البطيئة أو اكتشاف المشاكل المحتملة في الأداء.

Design Pattern ما هي:

تعريف: Design Patterns هي حلول عامة لمشاكل شائعة في تصميم البرمجيات، توفر هذه الأنماط قوالب جاهزة يمكن استخدامها لحل مشاكل معينة في البرمجة.

أنواع: تشمل الأنماط التصميمية الأساسية: Creational Patterns (مثل Singleton)، Structural Patterns (مثل Adapter)، و Behavioral Patterns (مثل Observer).

Joins أنواع:

INNER JOIN: يرجع فقط الصفوف التي لها تطابق في كلا الجدولين.

LEFT JOIN (or LEFT OUTER JOIN): يرجع كل الصفوف من الجدول الأيسر (الأول) بالإضافة إلى الصفوف المتطابقة من الجدول الأيمن.

RIGHT JOIN (or RIGHT OUTER JOIN): يرجع كل الصفوف من الجدول الأيمن (الثاني) بالإضافة إلى الصفوف المتطابقة من الجدول الأيسر.


FULL JOIN (or FULL OUTER JOIN): يرجع جميع الصفوف عندما يوجد تطابق في أحد الجدولين أو كليهما.

الفرق بين Class و Interface:

Class: هو قالب يحتوي على تعريف للبيانات (الخصائص) والسلوكيات (الدوال) الخاصة بكائنات معينة. يمكن إنشاء كائنات من الكلاس.

Interface: هو عقد يحدد مجموعة من الدوال بدون تنفيذها. الكلاسات التي تطبق الـ Interface يجب أن تنفذ هذه الدوال.

Independency Injection ما هو:

تعريف: هو نمط تصميم يهدف إلى فصل مكونات التطبيق  ن بعضها البعض، مما يسهل إدارتها واختبارها.

### الفرق بين View و Stored Procedure:

**Stored Procedure:** هي مجموعة من أوامر SQL المخزنة التي يمكن تنفيذها مباشرة في قاعدة البيانات.

**View:** هو استعلام SQL محفوظ يتم التعامل معه كجدول افتراضي لتجميع البيانات من جداول متعددة.  
**Indexed View ما هو:**

تعريف: هو View يحتوي على فهرس (Index) لتحسين أداء استعلامات معينة عن طريق تسريع عمليات البحث.

الاستخدام: يستخدم لتحسين أداء الاستعلامات التي تعتمد بشكل كبير على View معين.

### الفرق بين Clustered و Non-Clustered:

**Clustered Index:** يخزن البيانات الفعلية للجدول على القرص بالترتيب المحدد بواسطة الفهرس.

**Non-Clustered Index:** يخزن الفهرس فقط ولا يخزن البيانات الفعلية، بل يحتوي على مؤشر يشير إلى مواقع البيانات.

### الفرق بين For Loop و Foreach:

**For Loop:** يستخدم لتكرار عدد معين من المرات باستخدام فهرس يبدأ من قيمة وينتهي بقيمة أخرى.

**Foreach:** يستخدم لتكرار كل عنصر في مجموعة (مثل Array أو List) دون الحاجة إلى فهرس.

إذا كان لدي Row مع Id 10 وتم حذف هذا Row ثم قمت بإضافة Row جديد، ما هو Id الذي سيأخذه:

الجواب: إذا كان Id يتم توليده تلقائيًا، سيأخذ Row الجديد Id أعلى من آخر Id تم استخدامه، لذا لن يتم إعادة استخدام Id 10 بل سيتم إعطاء Id جديد، مثل 11 أو أعلى.

## Company 3

### الفرق بين Function و Stored Procedure:

**Stored Procedure:** يمكن أن يعيد أو لا يعيد نتيجة، ويمكن أن ينفذ عمليات مثل الإدراج، التحديث، أو الحذف.

**Function:** يعيد قيمة واحدة أو جدول بيانات، وعادةً ما يستخدم في الاستعلامات ولا يمكن أن يعدل البيانات بشكل مباشر.

إذا لم أكن أريد إرجاع بيانات مكررة، ماذا أستخدم:

**Distinct:** يستخدم لإزالة القيم المكررة في مجموعة النتائج.

كيف يمكنني معرفة أداء Join بين أكثر من جدول وهل الأ ↓ جيد أم لا:

**SQL Profiler:** يمكن استخدامه لتتبع أداء الاستعلامات ومقاس الزمن المستخدمة. قم بتنفيذ Join II.

البيانات بسجل ميسر.

إذا لم أكن أريد إرجاع بيانات مكررة، ماذا أستخدم:

Distinct: يستخدم لإزالة القيم المكررة في مجموعة النتائج.

كيف يمكنني معرفة أداء Join بين أكثر من جدول وهل الأداء جيد أم لا:

SQL Profiler: يمكن استخدامه لتتبع أداء الاستعلامات وقياس الزمن المستغرق في تنفيذ Join.

ما هو Temp Table في SQL:

تعريف: هو جدول مؤقت يتم إنشاؤه داخل جلسة عمل معينة لتخزين البيانات بشكل مؤقت.

OOP:

المفاهيم الأساسية: تشمل الوراثة (Inheritance)، التجريد (Abstraction)، التعددية (Polymorphism)، والتغليف (Encapsulation).

إذا كنت أريد نقل بيانات من Controller إلى View في MVC، ماذا أستخدم:

ViewBag: يتم استخدامه لنقل البيانات من Controller إلى View بشكل ديناميكي.

ViewData: مشابه لـ ViewBag لكنه يستخدم قواميس (Dictionary) لتخزين البيانات.

TempData: يستخدم لنقل البيانات بين طلبات مختلفة (مثلاً من Action إلى أخرى).

هل من الأفضل استخدام Use Static Files أم Use Routing في Middlewares:

Use Static Files: يستخدم لتقديم الملفات الثابتة مثل CSS و JavaScript.

Use Routing: يستخدم لتحديد المسارات (Routes) وتوجيه الطلبات إلى Controllers معينة.

الفرق بين Lazy Loading و Eager Loading:

Lazy Loading: يتم تحميل البيانات عند الحاجة فقط.

Eager Loading: يتم تحميل البيانات فوراً عند استدعاء الاستعلام.

الفرق بين Dependency Injection و Dependency Inversion:

Dependency Injection: هي تقنية يتم من خلالها حقن التبعية في الكائنات بدلاً من إنشائها داخلياً.

Dependency Inversion Principle (DIP): هو مبدأ تصميم يوصي بأن تعتمد الوحدات العليا في

التطبيق على التجريدات وليس على التنفيذات.

Design Pattern:

تعريف: هو حل عام قابل لإعادة الاستخدام لمشكلة شائعة في تصميم البرمجيات.

أنواع: تشمل Creational, Structural, Behavioral Patterns



ما هو Replication و Database Replica:



## :Company 4

الفرق بين Constructor و Destructor:

:Constructor

تعريف: هو دالة يتم استدعاؤها عند إنشاء كائن من الكلاس، يستخدم لتهيئة الكائن وإعداد قيمه الأولية.

الاستخدام: يتم استخدامه لتعيين القيم الابتدائية للخصائص وضمان أن الكائن في حالة صالحة للاستخدام.

:Destructor

تعريف: هو دالة يتم استدعاؤها عند تدمير كائن من الكلاس، يستخدم لتحرير الموارد التي تم تخصيصها للكائن.

الاستخدام: يتم استخدامه لتحرير الذاكرة والموارد الأخرى التي كان يستخدمها الكائن قبل تدميره.

ما هي Window Function:

تعريف: هي مجموعة من الدوال في SQL تستخدم لإجراء عمليات حسابية عبر مجموعة من الصفوف المرتبطة داخل جزء معين من الاستعلام.

الاستخدام: تُستخدم لعمليات مثل الترتيب، التجميع، والعمليات الحسابية على مجموعات من البيانات.

ما هو Class وما هو دوره:

تعريف: Class هو قالب أو مخطط يحدد الخصائص والسلوكيات (الدوال) لكائنات معينة.

الاستخدام: يتم استخدام Class لتجميع البيانات والسلوكيات التي تصف كائنًا معينًا في وحدة واحدة، مما يسهل تنظيم الكود.

أنواع Joins:

INNER JOIN: يرجع فقط الصفوف التي لها تطابق في كلا الجدولين.

LEFT JOIN (or LEFT OUTER JOIN): يرجع كل الصفوف من الجدول الأيسر (الأول) بالإضافة إلى الصفوف المتطابقة من الجدول الأيمن.

RIGHT JOIN (or RIGHT OUTER JOIN): يرجع كل الصفوف من الجدول الأيمن (الثاني) بالإضافة إلى الصفوف المتطابقة من الجدول الأيسر.

FULL JOIN (or FULL OUTER JOIN): يرجع جميع الصفوف عندما يوجد تطابق في أحد الجدولين أو كليهما.

أنواع Index:

**Clustered Index:** يخزن البيانات الفعلية للجدول على القرص بالترتيب المحدد بواسطة الفهرس.

**Non-Clustered Index:** يخزن الفهرس فقط ولا يخزن البيانات الفعلية، بل يحتوي على مؤشر يشير إلى مواقع البيانات.

الفرق بين Clustered Index و Non-Clustered Index:

**Clustered Index:** يخزن البيانات الفعلية للجدول بترتيب محدد، لذلك يمكن أن يكون هناك فقط Clustered Index واحد لكل جدول.

**Non-Clustered Index:** يخزن الفهرس فقط ويرتبط بمواقع البيانات الفعلية، ويمكن أن يكون هناك العديد من Non-Clustered Indexes لكل جدول.

SOLID:

تعريف: هي مجموعة من المبادئ الخمسة لتصميم البرمجيات التي تهدف إلى تحسين جودة الكود وجعله أكثر مرونة وقابلية للتوسع.

المبادئ:

**Single Responsibility Principle (SRP):** كل كلاس يجب أن يكون له مسؤولية واحدة فقط.

**Open/Closed Principle (OCP):** الكلاس يجب أن يكون مفتوحًا للتوسع ومغلقًا للتعديل.

**Liskov Substitution Principle (LSP):** الكائنات من النوع الفرعي يجب أن تكون قابلة للاستبدال بالكائنات من النوع الأساسي دون كسر الكود.

**Interface Segregation Principle (ISP):** يجب ألا يجبر المستخدمون على الاعتماد على واجهات لا يستخدمونها.

**Dependency Inversion Principle (DIP):** يجب أن تعتمد الوحدات العليا على التجريدات وليس على التفاصيل.

JWT (JSON Web Token)

تعريف: هو معيار مفتوح يستخدم لتبادل المعلومات بشكل آمن بين الأطراف المختلفة في صورة JSON.

الاستخدام: يستخدم بشكل واسع في أنظمة المصادقة لتأمين تبادل المعلومات بين العميل والخادم.

OOP Encapsulation:

تعريف: التغليف هو أحد مبادئ البرمجة الكائنية الذي يسمح بإخفاء تفاصيل تنفيذ الكود داخل الكائن وعدم السماح بالوصول إليها إلا من خلال واجهة معينة.



الاستخدام: يستخدم لحماية البيانات والتأكد من أن الكائنات تتفاعل فقط من خلال واجهات محددة مسبقًا.

:REST vs SOAP

REST: هو نمط معماري يستخدم بروتوكول HTTP ويعتمد على موارد يمكن الوصول إليها باستخدام URL.

SOAP: هو بروتوكول يستخدم لتبادل الرسائل بين الأنظمة عبر بروتوكولات متعددة، مثل HTTP وSMTP.

الفرق بين Stored Procedure و Function:

Stored Procedure: يمكن أن يعيد أو لا يعيد نتيجة، ويمكن أن ينفذ عمليات مثل الإدراج، التحديث، أو الحذف.

Function: يعيد قيمة واحدة أو جدول بيانات، وعادةً ما يستخدم في الاستعلامات ولا يمكن أن يعدل البيانات بشكل مباشر.

الفرق بين Trigger و Stored Procedure:

Stored Procedure: يتم استدعاؤها بشكل مباشر من قبل المستخدم أو التطبيق لتنفيذ مهمة معينة.

Trigger: يتم تنفيذه تلقائيًا استجابةً لأحداث معينة على جدول معين، مثل عمليات الإدراج أو التحديث أو الحذف.

:Execution Plan

تعريف: هو مخطط يتم إنشاؤه بواسطة قاعدة البيانات لعرض كيفية تنفيذ استعلام معين، ويعرض ترتيب تنفيذ العمليات مثل الفحص والمسح والفرز.

الاستخدام: يساعد في تحسين أداء الاستعلامات من خلال تحليل كيفية تنفيذها.

الفرق بين var و let في JavaScript:

var: يتم تعريفها على مستوى الدالة ويمكن إعادة تعريفها. لها نطاق واسع.


let: يتم تعريفها على مستوى الكتلة ولا يمكن إعادة تعريفها ضمن نفس النطاق. لها نطاق ضيق.

:Life Cycle of ASP.NET

تعريف: دورة حياة تطبيق ASP.NET تبدأ من استلام الطلب وتنتهي بإرسال الاستجابة إلى العميل.

المراحل: تشمل مراحل مثل الطلب (Request)، التوثيق (Authentication)، التوجيه (Routing)، معالجة الصفحة (Page Handling)، وإنشاء الاستجابة (Response Generation).

الفرق بين ASP.NET و MVC:

ASP.NET: هو إطار عمل لتطوير تطبيقات الويب يقدم الـ  من الأدوات والخدمات لتسهيل إنشاء التطبيقات.

ASP.NET: هو إطار عمل لتطوير تطبيقات الويب يقدم العديد من الأدوات والخدمات لتسهيل إنشاء التطبيقات.

MVC: هو نمط تصميم يتم استخدامه في تطوير الويب لفصل منطق التطبيق (Model)، العرض (View)، والتحكم (Controller) لتحسين تنظيم الكود.

الفرق بين Truncate و Delete و Drop:

Truncate: يزيل كل البيانات من الجدول دون حذف الجدول نفسه، ولا يمكن التراجع عنه.

Delete: يحذف الصفوف من الجدول بناءً على شرط معين ويمكن التراجع عنه إذا لم يتم التأكد.

Drop: يحذف الجدول نفسه من قاعدة البيانات بالكامل.

كيف يمكنني تحسين أداء SQL Query:

التحليل باستخدام Execution Plan: يمكن استخدام خطة التنفيذ لمعرفة كيفية تنفيذ الاستعلامات وتحليل الأجزاء التي تحتاج إلى تحسين.

فهرسة الجداول (Indexing): يمكن أن يساعد في تسريع الاستعلامات عن طريق تحسين عملية البحث.

تحسين الفلاتر (Filters): التأكد من أن الشروط والفلاتر تستخدم الفهارس بشكل صحيح.

الفرق بين Array و List:

Array: هو مجموعة ثابتة الحجم من العناصر من نفس النوع.

List: هو مجموعة ديناميكية يمكن أن يتغير حجمها، ويمكن أن تحتوي على عناصر من نفس النوع.

كيف يمكنني العثور على القيم المكررة في C#:

استخدام LINQ: يمكن استخدام استعلام LINQ للبحث عن القيم المكررة في مجموعة من البيانات.

GroupBy و Count: استخدام GroupBy لتجميع العناصر ثم استخدام Count لتحديد المجموعات التي تحتوي على أكثر من عنصر واحد.

الفرق بين Overloading و Overriding:

Overloading: هو تعريف دوال متعددة بنفس الاسم ولكن بمعاملات مختلفة داخل نفس الكلاس.

Overriding: هو تعديل دالة موروثية من كلاس أساسي في كلاس فرعي لتغيير سلوكها.

بهذا الشكل تكون الإجابات قد تم تنظيمها بالشكل الذي طلبته، من اليمين إلى اليسار، مع الحفاظ على التفاصيل والمعلومات كما هي.

## Company 5:

الفرق بين Authentication و Authorization:

Authentication (المصادقة):

تعريف: هي عملية التحقق من هوية المستخدم، أي تحديد من هو هذا المستخدم.  
الاستخدام: تستخدم لتأكيد أن الشخص الذي يحاول الوصول إلى النظام هو بالفعل من يدعي أنه هو.

Authorization (التفويض):

تعريف: هي عملية تحديد ما إذا كان المستخدم مصرحاً له بالقيام بعمليات معينة أو الوصول إلى موارد معينة.

الاستخدام: تأتي بعد المصادقة، وهي تحدد ما يمكن للمستخدم القيام به في النظام.

ما هي Serialization:

تعريف: هي عملية تحويل كائن إلى صيغة يمكن تخزينها أو نقلها (مثل JSON أو XML).  
الاستخدام: تستخدم لتبادل البيانات بين الأنظمة المختلفة أو لحفظ حالة الكائنات.

N-Tier Architecture:

تعريف: هو نمط تصميم معماري يتم فيه تقسيم التطبيق إلى طبقات متعددة (Tiers)، كل منها يخدم غرضًا معينًا ويعمل بشكل مستقل عن الطبقات الأخرى.  
الاستخدام: يتم استخدامه لتحسين تنظيم الكود، زيادة قابلية الصيانة، وتوزيع الأحمال عبر الخوادم.

OOP Encapsulation:

تعريف: التغليف هو أحد مبادئ البرمجة الكائنية الذي يسمح بإخفاء تفاصيل تنفيذ الكود داخل الكائن وعدم السماح بالوصول إليها إلا من خلال واجهة معينة.  
الاستخدام: يستخدم لحماية البيانات والتأكد من أن الكائنات تتفاعل فقط من خلال واجهات محددة مسبقًا.

JWT (JSON Web Token):

تعريف: هو معيار مفتوح يستخدم لتبادل المعلومات بشكل آمن بين الأطراف المختلفة في صورة JSON.



الاستخدام: يستخدم بشكل واسع في أنظمة المصادقة لتأمين تبادل المعلومات بين العميل والخادم.

.JSON

الاستخدام: يستخدم بشكل واسع في أنظمة المصادقة لتأمين تبادل المعلومات بين العميل والخادم.

:SOLID

تعريف: هي مجموعة من المبادئ الخمسة لتصميم البرمجيات التي تهدف إلى تحسين جودة الكود وجعله أكثر مرونة وقابلية للتوسع.

المبادئ:

Single Responsibility Principle (SRP): كل كلاس يجب أن يكون له مسؤولية واحدة فقط.

Open/Closed Principle (OCP): الكلاس يجب أن يكون مفتوحًا للتوسع ومغلقًا للتعديل.

Liskov Substitution Principle (LSP): الكائنات من النوع الفرعي يجب أن تكون قابلة للاستبدال بالكائنات من النوع الأساسي دون كسر الكود.

Interface Segregation Principle (ISP): يجب ألا يجبر المستخدمون على الاعتماد على واجهات لا يستخدمونها.

Dependency Inversion Principle (DIP): يجب أن تعتمد الوحدات العليا على التجريدات وليس على التفاصيل.

ما هو Route:

تعريف: Route هو المسار الذي يتم استخدامه لتوجيه الطلبات إلى الـ Controllers أو الإجراءات (Actions) المناسبة داخل التطبيق.

الاستخدام: يستخدم في تحديد كيفية التعامل مع طلبات HTTP وتوجيهها بناءً على عنوان URL.

الفرق بين Class و Struct و Interface:

:Class

تعريف: هو قالب يتم من خلاله إنشاء كائنات تحتوي على بيانات (خصائص) وسلوكيات (دوال).

:Struct

تعريف: هو نوع بيانات يشبه الكلاس لكنه يتم تعريفه عادة لتخزين البيانات البسيطة ويكون أخف من الكلاس.

:Interface

تعريف: هو عقد يحدد مجموعة من الدوال بدون تنفيذها. الكلاسات التي تطبق الـ Interface يجب أن تنفذ هذه الدوال.



الفرق بين Post و Get و Put و Delete:

Interface:

تعريف: هو عقد يحدد مجموعة من الدوال بدون تنفيذها. الكلاسات التي تطبق الـ Interface يجب أن تنفذ هذه الدوال.

الفرق بين Post و Get و Put و Delete:

Get: يستخدم لاسترجاع البيانات من السيرفر.

Post: يستخدم لإرسال البيانات إلى السيرفر لإنشاء مورد جديد.

Put: يستخدم لتحديث مورد موجود على السيرفر.

Delete: يستخدم لحذف مورد من السيرفر.

تحليل كود ومعرفة الأخطاء فيه:

الخطوة الأولى: قراءة الكود وفهم الوظيفة التي من المفترض أن يقوم بها.

الخطوة الثانية: التحقق من المنطق المستخدم في الكود ومطابقته مع متطلبات الوظيفة.

الخطوة الثالثة: فحص الكود بحثاً عن الأخطاء المحتملة مثل الأخطاء اللغوية، الأخطاء المنطقية، والأخطاء التي تتعلق بالتعامل مع البيانات.

شرح كود مع تحليل الوراثة (Inheritance):

Parent Class (Super Class): الكلاس الأساسي الذي يحتوي على الخصائص والدوال المشتركة.

Child Class (Sub Class): الكلاس الذي يرث الخصائص والدوال من الكلاس الأساسي ويمكنه إضافة خصائص ودوال جديدة أو تعديل الدوال الموروثة.

Overloading: هو تعريف دوال متعددة بنفس الاسم ولكن بمعاملات مختلفة داخل نفس الكلاس.

Overriding: هو تعديل دالة موروثة من كلاس أساسي في كلاس فرعي لتغيير سلوكها.

Virtual: يستخدم مع الدوال في الكلاس الأساسي للسماح للكلاس الفرعي بإعادة تعريف هذه الدوال.

New: يستخدم لإخفاء عضو موروثة من الكلاس الأساسي في الكلاس الفرعي.

الآن، النصوص مرتبة بشكل صحيح من اليمين إلى اليسار وفقاً لطلبك. إذا كنت بحاجة إلى مزيد من المعلومات أو تعديلات أخرى، فلا تتردد في طلب ذلك!

## 6:Company

التغليف (Encapsulation):

تعريف: التغليف هو أحد مبادئ البرمجة الكائنية الذي يسمح بإخفاء تفاصيل تنفيذ الكود داخل الكائن وعدم السماح بالوصول إليها إلا من خلال واجهة معينة.

الاستخدام: يستخدم لحماية البيانات والتأكد من أن الكائنات تتفاعل فقط من خلال واجهات محددة مسبقاً.

التجريد (Abstraction):

تعريف: التجريد هو عملية إخفاء التفاصيل المعقدة وإظهار الأشياء الأساسية فقط. يساعد في تقليل التعقيد من خلال التعامل مع الواجهات العامة بدلاً من التفاصيل.

الاستخدام: يستخدم لتحديد واجهات عامة للكائنات دون الكشف عن تفاصيل التنفيذ.

الفرق بين Overloading و Overriding:

Overloading: هو تعريف دوال متعددة بنفس الاسم ولكن بمعاملات مختلفة داخل نفس الكلاس.

Overriding: هو تعديل دالة موروثية من كلاس أساسي في كلاس فرعي لتغيير سلوكها.

محددات الوصول (Access Modifiers):

Public: يسمح بالوصول إلى العضو من أي مكان.

Private: يسمح بالوصول إلى العضو فقط من داخل نفس الكلاس.

Protected: يسمح بالوصول إلى العضو من داخل نفس الكلاس أو الكلاسات المشتقة.

Internal: يسمح بالوصول إلى العضو من داخل نفس المشروع فقط.

البرمجة الكائنية التوجه (OOP):

تعريف: هي نمط برمجي يعتمد على الكائنات (Objects) التي تحتوي على بيانات وسلوكيات.

المبادئ الأساسية: تشمل التغليف (Encapsulation)، التجريد (Abstraction)، التوريث (Inheritance)، والتعددية (Polymorphism).

مبادئ SOLID:

تعريف: هي مجموعة من المبادئ الخمسة لتصميم البرمجيات التي تهدف إلى تحسين جودة الكود وجعله أكثر مرونة وقابلية للتوسع.

المبادئ:

مبدأ المسؤولية الواحدة (Single Responsibility Principle - SRP): كل كلاس يجب أن يكون له



مسؤولية واحدة فقط.



مبدأ الفتح/الإغلاق (Open/Closed Principle - OCP): الكلاس يجب أن يكون مفتوحًا للتوسع ومغلقًا للتعديل.

مبدأ استبدال ليسكوف (Liskov Substitution Principle - LSP): الكائنات من النوع الفرعي يجب أن تكون قابلة للاستبدال بالكائنات من النوع الأساسي دون كسر الكود.

مبدأ فصل الواجهات (Interface Segregation Principle - ISP): يجب ألا يجبر المستخدمون على الاعتماد على واجهات لا يستخدمونها.

مبدأ عكس الاعتمادية (Dependency Inversion Principle - DIP): يجب أن تعتمد الوحدات العليا على التجريدات وليس على التفاصيل.

الفرق بين Singleton و Scoped و Transient:

Singleton: يتم إنشاء كائن واحد فقط يتم استخدامه طوال فترة تشغيل التطبيق.

Scoped: يتم إنشاء كائن جديد لكل طلب (Request) ويتم مشاركته خلال مدة الطلب.

Transient: يتم إنشاء كائن جديد في كل مرة يتم فيها طلب الخدمة.

الفرق بين Try و Catch و Finally:

Try: يحتوي على الكود الذي قد يؤدي إلى استثناءات (Exceptions).

Catch: يحتوي على الكود الذي يتم تنفيذه إذا حدث استثناء في بلوك Try.

Finally: يحتوي على الكود الذي يتم تنفيذه دائمًا بعد انتهاء تنفيذ Try و Catch، سواء حدث استثناء أم لا.

الفرق بين View و Function و Stored Procedure:

View: هو استعلام SQL محفوظ يمكن استدعاؤه كجدول افتراضي. يتم استخدامه لعرض البيانات.

Function: يعيد قيمة واحدة أو جدول بيانات، ويستخدم عادةً في الاستعلامات ولا يمكنه تعديل البيانات.

Stored Procedure: هي مجموعة من أوامر SQL المخزنة في قاعدة البيانات، يمكنها تنفيذ عمليات مثل الإدراج، التحديث، أو الحذف.

الفرق بين Boxing و Unboxing:

Boxing: هو تحويل نوع بيانات قيمة (مثل int) إلى كائن (Object).

Unboxing: هو تحويل كائن (Object) إلى نوع بيانات قيمة (مثل int).



مثال

ما هو Tracking في .NET Core:

تعريف: هو عملية تتبع التغييرات التي تحدث على الكائنات التي يتم جلبها من قاعدة البيانات باستخدام Entity Framework.

الاستخدام: يستخدم لتحديث البيانات في قاعدة البيانات عند استدعاء `SaveChanges()`.

الفرق بين Early Binding و Late Binding:

Early Binding: هو ربط الكائنات والدوال في وقت الترجمة (Compile Time).

Late Binding: هو ربط الكائنات والدوال في وقت التشغيل (Run Time)، ويستخدم بشكل عام مع الكائنات الديناميكية.

الفرق بين Struct و Class:

Struct: هو نوع بيانات يشبه الكلاس لكنه يتم تعريفه عادة لتخزين البيانات البسيطة ويكون أخف من الكلاس.

Class: هو قالب يتم من خلاله إنشاء كائنات تحتوي على بيانات (خصائص) وسلوكيات (دوال).

الفرق بين Mutable و Immutable:

Mutable: يعني أن الكائن يمكن تغيير حالته بعد إنشائه.

Immutable: يعني أن الكائن لا يمكن تغيير حالته بعد إنشائه.

الفرق بين Dispose و Finalize:

Dispose: يتم استخدامه لتحرير الموارد غير المُدارة بشكل صريح، وعادةً ما يُستدعى مباشرةً من قبل المطور.

Finalize: يتم استدعاؤه بواسطة جامع القمامة (Garbage Collector) لتحرير الموارد غير المُدارة قبل إزالة الكائن من الذاكرة.

## 7:Company

الفرق بين NET Core و .NET Framework:

.NET Core: هو إطار عمل مفتوح المصدر وعبر منصة (Cross-Platform) يستخدم لتطوير تطبيقات حديثة، يتميز بالأداء العالي والمرونة.

.NET Framework: هو إطار عمل قديم ومغلق المصدر تم تصميمه خصيصًا لتطوير تطبيقات ويندوز.

مبادئ SOLID:

تعريف: هي مجموعة من المبادئ الخمسة لتصميم البرمجيات التي تهدف إلى تحسين جودة الكود وجعله أكثر مرونة وقابلية للتوسع.

المبادئ:

مبدأ المسؤولية الواحدة (Single Responsibility Principle - SRP): كل كلاس يجب أن يكون له مسؤولية واحدة فقط.

مبدأ الفتح/الإغلاق (Open/Closed Principle - OCP): الكلاس يجب أن يكون مفتوحًا للتوسع ومغلقًا للتعديل.

مبدأ استبدال ليسكوف (Liskov Substitution Principle - LSP): الكائنات من النوع الفرعي يجب أن تكون قابلة للاستبدال بالكائنات من النوع الأساسي دون كسر الكود.

مبدأ فصل الواجهات (Interface Segregation Principle - ISP): يجب ألا يجبر المستخدمون على الاعتماد على واجهات لا يستخدمونها.

مبدأ عكس الاعتمادية (Dependency Inversion Principle - DIP): يجب أن تعتمد الوحدات العليا على التجريدات وليس على التفاصيل.

البرمجة الكائنية التوجه (OOP):

تعريف: هي نمط برمجي يعتمد على الكائنات (Objects) التي تحتوي على بيانات وسلوكيات.

المبادئ الأساسية: تشمل التغليف (Encapsulation)، التجريد (Abstraction)، التوريث (Inheritance)، والتعددية (Polymorphism).

Design Pattern (نمط التصميم):

تعريف: هو حل عام قابل لإعادة الاستخدام لمشكلة شائعة في تصميم البرمجيات.

الاستخدام: يستخدم لتسهيل عملية التصميم وزيادة قابلية الصيانة والتوسع.

#### :JWT (JSON Web Token)

تعريف: هو معيار مفتوح يستخدم لتبادل المعلومات بشكل آمن بين الأطراف المختلفة في صورة JSON.

الاستخدام: يستخدم بشكل واسع في أنظمة المصادقة لتأمين تبادل المعلومات بين العميل والخادم.  
:Oracle

تعريف: هو نظام إدارة قواعد بيانات علائقية (RDBMS) يستخدم لتخزين وإدارة البيانات. يتميز بقدراته العالية في معالجة البيانات وأدواته المتقدمة.

#### :SignalR

تعريف: هو مكتبة مفتوحة المصدر تستخدم لتسهيل عملية الاتصال في الوقت الحقيقي بين السيرفر والعميل في تطبيقات الويب.

الاستخدام: تستخدم لبناء تطبيقات تحتاج إلى تحديثات فورية مثل الدردشة، التنبيهات، أو الألعاب عبر الإنترنت.

#### :Repository with Unit of Work

تعريف: هو نمط تصميم يتم فيه استخدام الRepository كواجهة للتفاعل مع قاعدة البيانات، ويتم تنظيم العمليات في وحدة عمل (Unit of Work) لضمان أن جميع التغييرات تتم كمعاملة واحدة.  
الاستخدام: يستخدم لتحسين تنظيم الكود وتقليل الترابط بين طبقات التطبيق المختلفة.

#### :Plus

#### :DDD (Domain-Driven Design)

تعريف: هو منهجية تصميم تعتمد على التركيز على النطاق (Domain) الخاص بالمشروع وتحديد النماذج التي تعبر عنه بشكل دقيق.

الاستخدام: يستخدم لتبسيط عملية التصميم وتحقيق توافق أكبر بين الكود ومتطلبات العمل.

#### :UML (Unified Modeling Language)

تعريف: هي لغة نمذجة بصرية تستخدم لتصميم وتوثيق أنظمة البرمجيات.

الاستخدام: تستخدم لإنشاء مخططات تعبر عن بنية النظام، سلوكاته، وتفاعلاته.

#### :Clean Code

تعريف: هو مفهوم يشير إلى كتابة كود بسيط، واضح، ومنظم يسهل قراءته وفهمه.

الاستخدام: يهدف إلى تحسين جودة الكود وزيادة قابليته (↓)يانة والتوسع.

.....

## :Company 8

مبادئ البرمجة الكائنية (OOP Principles):

التغليف (Encapsulation): إخفاء تفاصيل تنفيذ الكود داخل الكائن وعدم السماح بالوصول إليها إلا من خلال واجهة معينة.

التجريد (Abstraction): إخفاء التفاصيل المعقدة وإظهار الجوانب الأساسية فقط.

التوريث (Inheritance): إمكانية إنشاء كلاس جديد يعتمد على خصائص وسلوكيات كلاس آخر.

التعددية (Polymorphism): قدرة الكائنات على استخدام الواجهات نفسها لتنفيذ سلوكيات مختلفة.

مبادئ SOLID:

مبدأ المسؤولية الواحدة (Single Responsibility Principle - SRP): كل كلاس يجب أن يكون له مسؤولية واحدة فقط.

مبدأ الفتح/الإغلاق (Open/Closed Principle - OCP): الكلاس يجب أن يكون مفتوحًا للتوسع ومغلقًا للتعديل.

مبدأ استبدال ليسكوف (Liskov Substitution Principle - LSP): الكائنات من النوع الفرعي يجب أن تكون قابلة للاستبدال بالكائنات من النوع الأساسي دون كسر الكود.

مبدأ فصل الواجهات (Interface Segregation Principle - ISP): يجب ألا يجبر المستخدمون على الاعتماد على واجهات لا يستخدمونها.

مبدأ عكس الاعتمادية (Dependency Inversion Principle - DIP): يجب أن تعتمد الوحدات العليا على التجريدات وليس على التفاصيل.

الفرق بين .NET Framework و .NET Core:

.NET Framework: إطار عمل مغلق المصدر تم تصميمه بشكل رئيسي لتطوير تطبيقات ويندوز.

.NET Core: إطار عمل مفتوح المصدر وعبر منصة (Cross-Platform) يستخدم لتطوير تطبيقات حديثة ومتعددة الأنظمة.

الفرق بين Abstract و Interface:

Abstract: كلاس يمكن أن يحتوي على دوال مفعلة وغير مفعلة، ويتم استخدامه لتوفير قاعدة يمكن للكلاسات المشتقة البناء عليها.

Interface: عقد يحدد مجموعة من الدوال بدون تنفيذها. الكلاسات التي تطبق الـ Interface يجب أن تنفذ هذه الدوال.



الفرق بين Function و Stored Procedure:

Function: يعيد قيمة واحدة أو جدول بيانات، ويستخدم عادةً في الاستعلامات ولا يمكنه تعديل البيانات.

Stored Procedure: هي مجموعة من أوامر SQL المخزنة في قاعدة البيانات، يمكنها تنفيذ عمليات مثل الإدراج، التحديث، أو الحذف.

الفرق بين Clustered Index و Non-Clustered Index:

Clustered Index: يخزن البيانات الفعلية للجدول بترتيب محدد، ويمكن أن يكون هناك Clustered Index واحد فقط لكل جدول.

Non-Clustered Index: يخزن الفهرس فقط ولا يخزن البيانات الفعلية، ويمكن أن يكون هناك العديد من Non-Clustered Indexes لكل جدول.

استعلامات SQL:

تعريف: هي أوامر تستخدم للتفاعل مع قواعد البيانات بهدف استرجاع أو تعديل البيانات.

الاستخدام: تشمل الاستعلامات مثل SELECT، INSERT، UPDATE، DELETE.

الفرق بين ViewBag و ViewData و TempData:

ViewBag: يستخدم لنقل البيانات من الـ Controller إلى الـ View بشكل ديناميكي.

ViewData: مشابه لـ ViewBag لكنه يستخدم القواميس (Dictionary) لتخزين البيانات.

TempData: يستخدم لنقل البيانات بين طلبات مختلفة (Actions) في نفس الجلسة.

الفرق بين AddScoped و Singleton و Transient:

AddScoped: يتم إنشاء كائن جديد لكل طلب (Request) ويتم مشاركته خلال مدة الطلب.

Singleton: يتم إنشاء كائن واحد فقط يتم استخدامه طوال فترة تشغيل التطبيق.

Transient: يتم إنشاء كائن جديد في كل مرة يتم فيها طلب الخدمة.

نمط التصميم (Design Pattern):

تعريف: هو حل عام قابل لإعادة الاستخدام لمشكلة شائعة في تصميم البرمجيات.

الاستخدام: يستخدم لتسهيل عملية التصميم وزيادة قابلية الصيانة والتوسع.

الهندسة ثلاثية الطبقات (3-Tier Architecture):

تعريف: هو نمط تصميم معماري يتم فيه تقسيم التطبيق إلى ثلاث طبقات: الطبقة التقديمية (Presentation Tier)، الطبقة المنطقية (Logic Tier)، وطبقة البيانات (Data Tier).

الاستخدام: يستخدم لتحسين تنظيم الكود وزيادة مرونة التطبيق والتطوير.

### الهندسة متعددة الطبقات (N-Tier Architecture):

تعريف: هو نمط تصميم معماري يتم فيه تقسيم التطبيق إلى أكثر من ثلاث طبقات، حيث تخدم كل طبقة غرضًا معينًا.

الاستخدام: يستخدم لتنظيم التطبيقات الكبيرة والمعقدة وزيادة قابليتها للتوسع.

### الهندسة البصلية (Onion Architecture):

تعريف: هو نمط تصميم معماري يهدف إلى فصل منطق الأعمال عن البنية التحتية من خلال ترتيب الكود في طبقات تشبه الطبقات البصلية.

الاستخدام: يستخدم لتسهيل اختبار الكود وتعزيز مرونة التطوير والصيانة.

### نمط (MVC Pattern) MVC:

تعريف: هو نمط تصميم يستخدم لفصل منطق التطبيق (Model)، العرض (View)، والتحكم (Controller) لتحسين تنظيم الكود.

الاستخدام: يستخدم على نطاق واسع في تطوير تطبيقات الويب لتوفير هيكل تنظيمي واضح وسهل الصيانة.

### JWT (JSON Web Token):

تعريف: هو معيار مفتوح يستخدم لتبادل المعلومات بشكل آمن بين الأطراف المختلفة في صورة JSON.

الاستخدام: يستخدم بشكل واسع في أنظمة المصادقة لتأمين تبادل المعلومات بين العميل والخادم.

### Middleware:

تعريف: هو مكون في ASP.NET Core يتعامل مع طلبات HTTP قبل أن تصل إلى الـ Controller وبعد أن تخرج منه.

الاستخدام: يستخدم لإجراء عمليات مثل المصادقة، تسجيل الطلبات، التوجيه، ومعالجة الأخطاء.


### Pipeline:

تعريف: هو سلسلة من الـ Middleware في ASP.NET Core التي يتم من خلالها تمرير الطلبات والاستجابات.

الاستخدام: يستخدم لتحديد كيف يتم معالجة الطلبات من بداية استلامها حتى إرسال الاستجابة.

### Multithreading (تعدد الخيوط):

تعريف: هو تقنية تسمح بتنفيذ مهام متعددة في نفس الوقت في نفس العملية.

الاستخدام: يستخدم لتحسين أداء التطبيقات وزيادة كفاءتها  خلال توزيع المهام على خيوط متعددة.

تعريف: هو تقنية تسمح بتنفيذ مهام متعددة في نفس الوقت في نفس العملية.  
الاستخدام: يستخدم لتحسين أداء التطبيقات وزيادة كفاءتها من خلال توزيع المهام على خيوط متعددة.

الفرق بين View و Stored Procedure:

View: هو استعلام SQL محفوظ يمكن استدعاؤه كجدول افتراضي. يتم استخدامه لعرض البيانات.  
Stored Procedure: هي مجموعة من أوامر SQL المخزنة في قاعدة البيانات، يمكنها تنفيذ عمليات مثل الإدراج، التحديث، أو الحذف.

الفرق بين المفتاح الأساسي (Primary Key) والمفتاح المركب (Composite Key):

Primary Key: هو عمود أو مجموعة من الأعمدة التي تميز كل صف في الجدول بشكل فريد.  
Composite Key: هو مفتاح يتألف من عمودين أو أكثر، يتم استخدامه لتحديد كل صف في الجدول بشكل فريد.

المؤشر (Indexer):

تعريف: هو عضو خاص في الكلاس يسمح بالوصول إلى عناصر الكائنات باستخدام فهرس مشابهة لمصفوفة.

الاستخدام: يستخدم لتوفير طريقة مريحة للوصول إلى عناصر الكائنات.

Join (الانضمام):

تعريف: هو عملية دمج بيانات من جداول متعددة بناءً على شرط معين.

أنواع:

INNER JOIN: يرجع الصفوف التي لها تطابق في كلا الجدولين.

LEFT JOIN: يرجع جميع الصفوف من الجدول الأيسر بالإضافة إلى الصفوف المتطابقة من الجدول الأيمن.

RIGHT JOIN: يرجع جميع الصفوف من الجدول الأيمن بالإضافة إلى الصفوف المتطابقة من الجدول الأيسر.

FULL JOIN: يرجع جميع الصفوف من كلا الجدولين، سواء كان هناك تطابق أم لا.

الآن، النصوص مرتبة بشكل صحيح من اليمين إلى اليسار، إذا كنت بحاجة إلى المزيد من المساعدة أو التوضيحات، فأنا هنا للمساعدة!



## :Company 9

ما هو مبدأ تصميم (IOC (Inversion of Control:

تعريف: مبدأ تصميم يعكس التحكم في تدفق البرنامج من الكود الخاص بك إلى إطار العمل أو المكتبة، حيث يقوم إطار العمل بإدارة إنشاء الكائنات والاعتماديات بدلاً من أن يقوم الكود الخاص بك بذلك.

الاستخدام: يستخدم لتحسين مرونة الكود وتقليل الترابط بين مكونات النظام.

ما هو إطار عمل (IOC Framework) (IOC):

تعريف: هو إطار عمل يسهل تطبيق مبدأ انعكاس التحكم (IOC) من خلال إدارة الاعتماديات بين الكائنات.

أمثلة: Spring Framework و.NET Core Dependency Injection هما أمثلة على أطر عمل IOC.

الفرق بين مبادئ التصميم (Design Principles) وأنماط التصميم (Design Patterns):

مبادئ التصميم (Design Principles): هي مجموعة من الإرشادات العامة التي تهدف إلى تحسين جودة البرمجيات، مثل مبادئ SOLID.

أنماط التصميم (Design Patterns): هي حلول جاهزة لمشاكل شائعة في تصميم البرمجيات، مثل Singleton وFactory Pattern.

ما هي علاقات البرمجة الكائنية التوجه (OOP Relationships):

تعريف: العلاقات بين الكائنات في البرمجة الكائنية تشمل التوريث (Inheritance)، التركيب (Composition)، والتجميع (Aggregation).

الاستخدام: تساعد في تنظيم الكود وجعل العلاقات بين الكائنات أكثر وضوحاً.

الفرق بين علاقات Is A و Has A:

Is A: تعبر عن علاقة توريث (Inheritance) حيث يكون الكائن هو نوع فرعي من نوع آخر.

Has A: تعبر عن علاقة تكوين (Composition) حيث يحتوي الكائن على كائن آخر كجزء منه.

شرح التغليف (Encapsulation) في البرمجة:

تعريف: التغليف هو مبدأ في البرمجة الكائنية يقوم بإخفاء تفاصيل تنفيذ الكود داخل الكائن وعدم السماح بالوصول إليها إلا من خلال واجهة معينة.

الاستخدام: يهدف إلى حماية البيانات وتسهيل صيانة الكود من خلال توفير واجهات واضحة للتفاعل مع الكائنات.



ما هي Signature للMethod:

تعريف: هي تعريف دالة معينة يتضمن اسم الدالة، نوع البيانات التي تستقبلها، ونوع البيانات التي تعيدها.

الاستخدام: يساعد في تمييز الدوال عن بعضها البعض خاصة عند استخدام Overloading.

ما هو Use Case وما علاقته بStatic Constructor:

تعريف Use Case: هو وصف لمجموعة من التفاعلات بين المستخدم والنظام لتحقيق هدف معين.

Static Constructor: هو نوع من Constructors في C# يتم استدعاؤه مرة واحدة فقط عندما يتم تحميل الكلاس لأول مرة، ويستخدم عادةً لتهيئة البيانات الثابتة.

متى يتم استخدام Composition بدلاً من Inheritance:

التركيب (Composition): يتم استخدامه عندما نريد تجميع سلوكيات متعددة في كائن واحد دون الحاجة إلى استخدام التوريث.

التوريث (Inheritance): يتم استخدامه عندما نريد إعادة استخدام سلوكيات كلاس موجود في كلاس جديد.

الحل الأمثل: إذا كان هناك حاجة لإعادة استخدام كود مشترك بين كائنات متعددة دون إنشاء تسلسل هرمي معقد، فإن Composition يكون الخيار الأفضل.

ما هي الـ Members في الـ Class وكيف تختلف عن الـ Members القابلة للتوريث:

Members في الـ Class: هي الخصائص والدوال المعرفة داخل الكلاس.

Members القابلة للتوريث: هي الخصائص والدوال التي يمكن للكلاس الفرعي أن يرثها ويستخدمها أو يعيد تعريفها.

هل C# تعتبر لغة Strongly Typed أو Loosely Typed:

Strongly Typed: تعني أن نوع البيانات يجب أن يتم تحديده بوضوح في وقت الترجمة ولا يمكن تغييره بسهولة.

Loosely Typed: تعني أن نوع البيانات يمكن أن يتغير في وقت التشغيل.

C#: تعتبر لغة Strongly Typed.

ما هو الفرق بين Static Polymorphism و Dynamic Polymorphism:

Static Polymorphism (التعددية الثابتة): يتم تحقيقها باستخدام Overloading، حيث يتم تحديد الدالة التي سيتم استدعاؤها في وقت الترجمة.

Dynamic Polymorphism (التعددية الديناميكية): يتم تحقيقها باستخدام Overriding، حيث يتم تحديد الدالة التي سيتم استدعاؤها في وقت التشغيل.

الفرق بين Value Type و Reference Type:

Value Type: تخزن القيم مباشرة في الذاكرة، مثل int و bool.

Reference Type: تخزن مراجع للبيانات المخزنة في مواقع الذاكرة، مثل الكائنات والمصفوفات.

الفرق بين Hashtable و Dictionary:

Hashtable: يخزن البيانات كأزواج مفتاح/قيمة ويستخدم للتخزين غير المترابط. يدعم أنواع البيانات المرجعية فقط.

Dictionary: يشبه Hashtable لكنه يتميز بأنه Generic، أي يمكنه التعامل مع أنواع بيانات محددة.

الفرق بين Primary Key و Unique Key في SQL:

Primary Key: يميز كل صف في الجدول بشكل فريد ولا يمكن أن يحتوي على قيم فارغة.

Unique Key: يضمن أن جميع القيم في العمود فريدة لكنه يمكن أن يحتوي على قيم فارغة.

الفرق بين var و dynamic:

var: يتم تحديد نوع المتغير في وقت الترجمة ولا يمكن تغييره.

dynamic: يتم تحديد نوع المتغير في وقت التشغيل ويمكن تغييره.