سوال 2:

بخش 1:

در Inner join فقط رکوردهایی نمایش داده میشوند که در شرط مطابقت داشته باشند.و اگر در جدول اول رکوردی وجود داشته باشه که با هیچ رکوردی از جدول دوم مطابقت نداشته باشه، در نتیجه نهایی نمایش داده نمیشه.

در Left join تمام رکوردها از جدول اول(جدول سمت چپ join) نمایش داده میشه، حتی اگر با هیچ رکوردی از جدول دوم مطابقت نداشته باشه.

Right join برعکس left join عمل میکنه و تمام رکوردهای جدول دوم(جدول سمت راست join) رو نمایش میده، حتی اگر با هم مطابقت نداشته باشند

بخش 2:

ایندکس باعث می شود کوئری ها سریع شود" این ساده ترین تعریف از ایندکس ها هست.

با استفاده از عبارت ‍create index‍ میتوانیم ایندکس بسازیم. با ساخت ایندکس یک کپی از داده ها در جای دیگر ساخته می شود که مانند فهرست یک کتاب به داده ها در داخل جدول اصلی اشاره دارند. توجه داشته باشید که هر ایندکس فضای ذخیره سازی مخصوص خودش را لازم دارد.

جست و جو با کمک ایندکس مانند پیدا کردن شماره ای در دفتر تلفن است که با توجه به مرتب بودن شماره ها خیلی سریع شماره ها رو پیدا میکنیم اما ایندکس ها در دیتابیس پیچیده تر از یک دفترچه تلفن هستند زیرا به طور مداوم داده ها در حال تغییر (update, delete, insert) هستند و با این حال دیتابیس باید ایندکس ها رو مرتب نگه دارد.

دیتابیس از دو ساختار برای پاسخگویی به این چالش استفاده می کند:

doubly linked list

The Search Tree (B-Tree)

این دو ساختار ، بیشتر ویژگیهای عملکرد دیتابیس را توضیح می دهند.

هدف اولیه هر ایندکس فراهم آوردن یک نمایش منظم از داده‌های ایندکس است. ذخیره داده ها به صورت متوالی امکان پذیر نیست زیرا یک عبارت درج باید داده ها را جابجا کند تا فضای جدیدی را فراهم کند. انتقال مقادیر زیادی از داده بسیار زمان بر است بنابراین کوئری insert بسیار کند می شود.

راه حل مسئله ایجاد نظمی منطقی است که مستقل از نظم فیزیکی در حافظه باشد.

بخش 3:

منظور از Subquery یک query است که درون یک عبارت مانند UPDATE، INSERT، SELECT یا DELETE نوشته می‌شود.

انواع Subquery در SQL Server

می‌توانید در جاهای زیادی از یک subquery استفاده کنید:

در محل یک عبارت

همراه با IN یا NOTIN

همراه با ANY یا ALL

همراه با EXISTS یا NOT EXISTS

در کدهای DELETE، UPDATE یا INSERT

بخش 4:

نرمال سازی فرآنید و یا تیکنیکی است در طراحی یا طراحی مجدد یک پایگاه داده، جهت بهینه سازی و حذف افزونگی داده ها و تضمین این امر در فرم های مختلف نرمال سازی.

پایگاه داده ای که نرمال سازی نشده است، ممکن است شامل داده ای باشد که بدون دلیل خاصی در جداول بانک اطلاعاتی آن به صورت تکراری موجود و ذخیره سازی شده است، که این امر میتواند  بانک اطلاعاتی را در مواردی چون : مصرف حافظه بانک اطلاعاتی، سرعت اجرای پرس و جو ها، بازدهی بروزرسانی پایگاه داده و مهم تر از همه تمامیت و درستی داده ها را دچار مشکل سازد.

بانک اطلاعاتی قبل از نرمال سازی متشکل از واحد های غیر منطقی و جداول مدیریت نشده ای است که در اصطلاح آن را پایگاه داده خام می نامند.

طراحی و نرمال سازی بر اساس نیاز های کاربر:

فراموش نکنید که نیاز های کاربر باید همواره محور اصلی طراحی یک پایگاه داده واقع شود و نباید این نکته را نیز فراموش کرد که کاربر پایگاه داده بیشترین استفاده را از این سیستم داراست پس باید در نهایت در رابط کاربری برای وی بیشترین سادگی و راحتی فراهم و مهیا شود و این امر تحقق نخواهد یافت مگر آنکه در کنار طراحی نرمال سازی و بهینه شده ، نیاز های کاربر مورد بررسی قرار گرفته باشند

چه داده هایی باید در جهت نیاز های کاربر در بانک اطلاعاتی ذخیره گردند؟

چگونه کاربر به بانک اطلاعاتی دسترسی دارد؟

چه دسترسی هایی کاربر نیاز دارد؟

چگونه داده های را در پایگاه داده باید گروه بندی کرد؟

چه داده ای بیشتر مورد دسترسی قرار میگیرد؟

نحوه ارتباط داده ها در بانک اطلاعاتی چگونه است؟

چه محاسباتی جهت ارائه دقیق داده باید انجام گیرد؟

فواید نرمال سازی :

فرآیند نرمال سازی فواید بسیاری را برای پایگاه داده در بر خواهد داشت که  از جمله این فواید میتوان موارد زیر را نام برد:

سازمان دهی یکپارچه بانک اطلاعاتی

سازگاری داده در پایگاه داده

طراحی منعطف تر پایگاه داده

مدیریت بهتر پایگاه داده از لحاظ امنیتی

فرآیند Denormalization چیست؟

در سطوح بالای نرمال سازی، گاهی کاهش افزونگی داده ها معایبی را نیز به دنبال خواهند داشت. مثلا فرآیند نرمال سازی باعث کاهش زمان پاسخگویی می شود و یا در پرس و جو های مختلف اغلب باید همه ی جداول بوجود آمده در فرم نرمال با یکدیگر الحاق شوند تا پرس و جو صورت گیرد، که زمان زیادی را سپری خواهد کرد و در سرویس های پایگاه داده وب که زمان پاسخ گویی از فاکتور های مهم است، اصلا مناسب نیست. در این نوع پایگاه داده ها که در آنها افزونگی داده اهمیت چندانی ندارد میتوان سطح نرمال سازی را کاهش داد که این فرآیند را Denormalization گویند.

بخش 5:

تئوری پايگاه داده درجه نرمال سازی جدول را با اصطلاح فرم های نرمال (normal form) شرح می دهد. فرم های نرمال (يا بطور خلاصه NF) معياری برای تعيين درجه نرمال جدول دراختيار می گذارد. فرم های نرمال جداگانه روی هر جدول می توانند بکار بروند. پايگاه داده زمانی در فرم نرمال n خواهد بود که کل جداول آن در فرم نرمال n باشند.

نرمال سازی First Normal Form) 1NF)

يک جدول در فرم اول نرمال (1NF) است اگر و فقط اگر فاقد گروه داده تکرار شونده باشد. به عبارت ديگر هر ستون در جدول دارای مقدار اتميک باشد. در مدل رابطه ای هر جدولی حداقل در فرم اول نرمال هست زيرا از الزامات مدل اين است که هر جدول شامل دقيقا يک مقدار برای هر صفت خاصه باشد که اصطلاحا “فاقد گروه تکرار شونده” گفته می شود.

نرمال سازی Second Normal Form) 2NF)

يک جدول در فرم دوم نرمال (2NF) است اگر اولا 1NF باشد و ثانيا کليه ستون های غيرکليد با کليد اصلی وابستگی تابعی کامل داشته باشند. ستون Y با ستون X در يک رابطه وابستگی تایعی (functional dependency) دارد اگروفقط اگر به ازای هر مقدار در X دقيقا يک مقدار در Y متناظر با آن وجود داشته باشد. که به صورت X→Y نشان داده می شود.

نرمال سازی Third Normal Form) 3NF)

يک جدول در فرم سوم نرمال (3NF) است اگر اولا 2NF باشد، ثانيا کليه صفات خاصه غير کليد در جدول با کليد اصلی وابستگی تابعی غير تعدی داشته باشند. وابستگی تعدی (transitive dependency) يک وابستگی تابعی غير مستقيم است که در آن X→Z است اگر X→Y و Y→Z باشد.

نمرالسازی به فرم BCNF

رابطه R در سطح BCNF است اگر و فقط اگر هر دترمینان، کلید کاندید باشد به عبارتی  دیگرجدولی در BCNF است که ستون های آن فقط به کلید های کاندیدش وابستگی تابعی  داشته باشند.

هر صفت خاصه ای که صفت خاصه ای دیگر با آن وابستگی تابعی کامل داشته باشد، دترمینان  نامیده می شود. یعنی اگر داشته باشیم

 R. A -> R. B

 و این وابستگی کامل باشد صفت خاصه A را دترمینان می نامیم

بخش 6:

با TRUNCATE می توانید به طور کامل رکوردهای جدول را از بین برده و فضای اشغال شده توسط رکوردهای جدول را آزاد کنید.

Syntax آن به فرم زیر می باشد:

TRUNCATE TABLE table\_name;

در TRUNCATE اولا همه رکوردها یک جدول حذف شده و فضای آن جدول هم آزاد می شود وضمنا کلیه ستون های identity ریست می شوند و انکار در این جدول هیچ رکوردی قبلا اضافه نشده بوده است.

دستور DROP برای حذف یک شیء از دیتابیس مورد استفاده قرار می گیرد. اگر جدولی را از دیتابیس DROP کنید کلیه رکوردها و ساختار آن جدول از دیتابیس از بین می رود. وقتی جدولی DROP شود، برگرداندن آن غیرممکن است مگر آنکه بک آپ داشته باشید. پس در زمان استفاده از دستور DROP بسیار مراقب باشید(خصوصا در محیط واقعی).

Syntax استفاده از دستور DROP به فرم زیر می باشد.

DROP TABLE table\_name;

با drop شدن یک جدول، کلیه ارتباطات و Reference های آن با جداول دیگر و از جداول دیگر با این جدول، بی اعتبار خواهد شد. کلیه دسترسی های تعیین شده به این جدول از بین خواهد رفت. اگر می خواهید دوباره از جدول استفاده کنید ، باید با آن دوباره ایجاد کنید ضمن اینکه سایر مواردی که توضیح داده شد هم باید دوباره دستی ایجاد کنید. ولی وقتی جدولی را truncate کنید همه چیز به غیر از رکوردهای جدول سرجای خود باقی خواهد ماند.