**C# code Question solution:**

בשאלה 1 הבעיה בפונקציה InculdeInFinalResults היא בפעולה ה- switch

כאשר רוצים לבדוק את הערך של ה-enum צריך לעשות בצורה הנ"ל

public bool IncludeInFinalResults(Room room){

bool isAvailable = CheckIfRoomAvailable(room);

switch (enforcedOption)

{

case RoomAvailableOption.All: return true;

case RoomAvailableOption.AvailableOnly: return isAvailable;

case RoomAvailableOption.NotAvailableOnly: return !isAvailable;

default: return true;

}

}

יש אופציה נוספת שבו אפשר , הערכים הדיפולטיבים של האופציות ב-enum זה מאפס והבא גדל באחד שוב ושוב   
כך ש:

public enum RoomAcailableOption{

All (is number 0),

AvailableOnly (is number 1),

NotAvailableOnly (is number 2)

}

ובכך אפשר לשנות את הקוד שזה יתאים למטרה של ה-switch

public bool IncludeInFinalResults(Room room){

bool isAvailable = CheckIfRoomAvailable(room);

switch ((int)enforcedOption)

{

case 0: return true;

case 1: return isAvailable;

case 2: return !isAvailable;

default: return true;

}}

פתרון לשאלה השניה הוא

בפתרון זה אפשר לראות שהוא יותר קריא מכיוון שכל אחת מהעמודות ניגש לפונקציה הבודקת לגבי ה-null גם בdata Base וגם ב- .net null.

private Object dataUser(DataRow dr,string columnName){

if(dr[columnName]!=DBNull.Value && dr[columnName]!=null) {

return dr[columnName].ToString();

}

return null;

}

public User GetUserFromDB(int userId){

User user = new User();

user.UserId = userId;

DataRow dr = DataRepository.GetUserById(userId);

user.FirstName = dataUser(dr, "FirstName");

user.LastName = dataUser(dr, "LastName");

user.Address = dataUser(dr, "Address");

user.CityName = dataUser(dr, "CityName");

user.CountryName = dataUser(dr, "CountryName");

user.Email = dataUser(dr,"Email");

return user;

}

בנוסף, במידה ונרצה לעשות זאת בצורה כללית כמו גינרית (ריפלקשין – אבל יש בה גם חיסרון שהוא תופס הרבה מהזיכרון וכדי לעשות זאת רק במידה ולא יודעים את המאפינים של התא או שיוצרים אותם תיכנותית)

public User GetUserFromDB(int userId)

{

User user = new User();

user.UserId = userId;

DataRow dr = DataRepository.GetUserById(userId);

Type objectType = typeof(User);

//return all the properties the user have like userId,firstName and more

foreach(ProperyInfo property in objectType.GetProperties()){

string propertyName = property.Name;

//its can be that the user has proparty that the data row has not

if (dr.Table.Columns.Contains(propertyName)){

if (dr[propertyName] != DBNull.Value && dr[propertyName] != null){

property.SetValue(user,dr[propertyName].ToString());

}

}

}

return user;}

נקודה חשובה בפתרון הזה הוא מסתמך על זה שה-DATA ROW מקושר ל-DATA TABLE אם הוא לא מקושר צריך לעשות מערך של כל הcolumns שנמצאים ב-data row ולבדוק אם הרשימה מכילה את הproparty הזה

string[] arr = new string[]{"FirstName","LastName","Address","CityName","CountryName","Email"};  
ולבדוק אם הוא מכיל פה

**Sql Question Solution:**

בשאלה הראשונה הפתרון הוא:

select BANK\_ID,BRANCH\_ID,ACCOUNT\_NUM,CURRENCY,AMOUNT

from ACCOUNTS join AMOUNTS on concat\_ws(' ',BANK\_ID,BRANCH\_ID,ACCOUNT\_NUM) = ACCOUNT\_REC

בשאלה השניה הפתרון הוא:

הפעולה שלוקחת הרבה זמן היא בכך שבכל שורה הקיימת בטבלה אנו ניפגשים לפונקצית עזר CALC\_AMT ובכך כל פעם אנו עושים את החישוב הזה.  
כדי להפחית את זמן הריצה אנו צריכים למצוא דרך בה אפשר להוריד את זמן הריצה