בדוח הזה נתאר את השילוב של המערכת שלנו עם מערכת של זוג אחר  
נתאר את ההחלטות והשינויים שאנחנו נבצע במערכת הקיימת שלנו.

A screen shot of a computer program

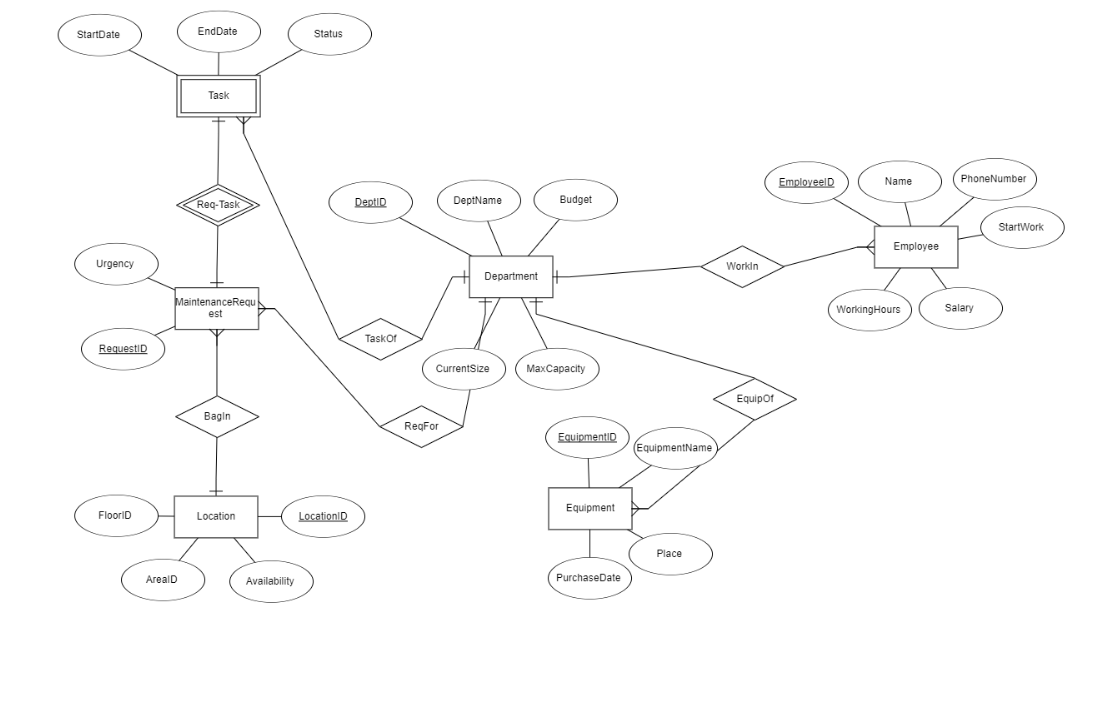
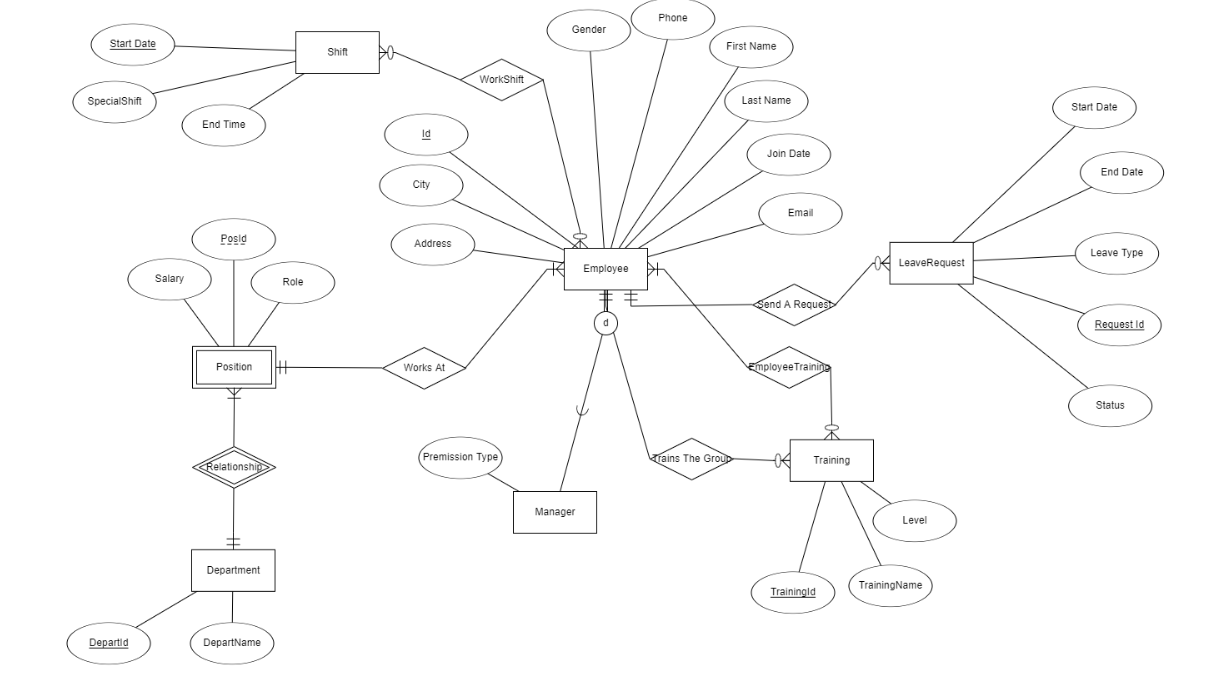
Description automatically generated

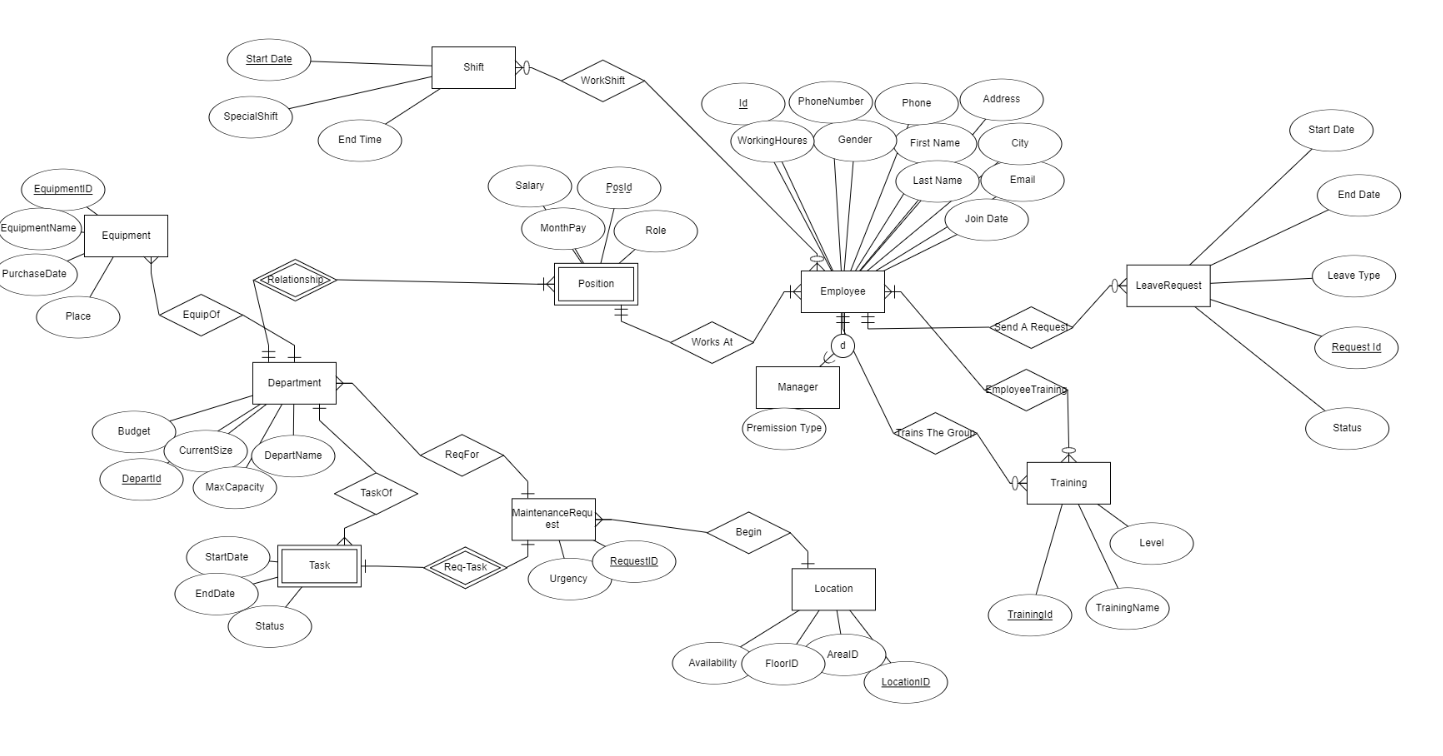
הזוג הביא לנו קובץ גיבוי שנעשה בשיטת SQL נחלץ את תיאור הטבלאות על מנת לבנות את דיאגרמות ה-DSD וה-ERD

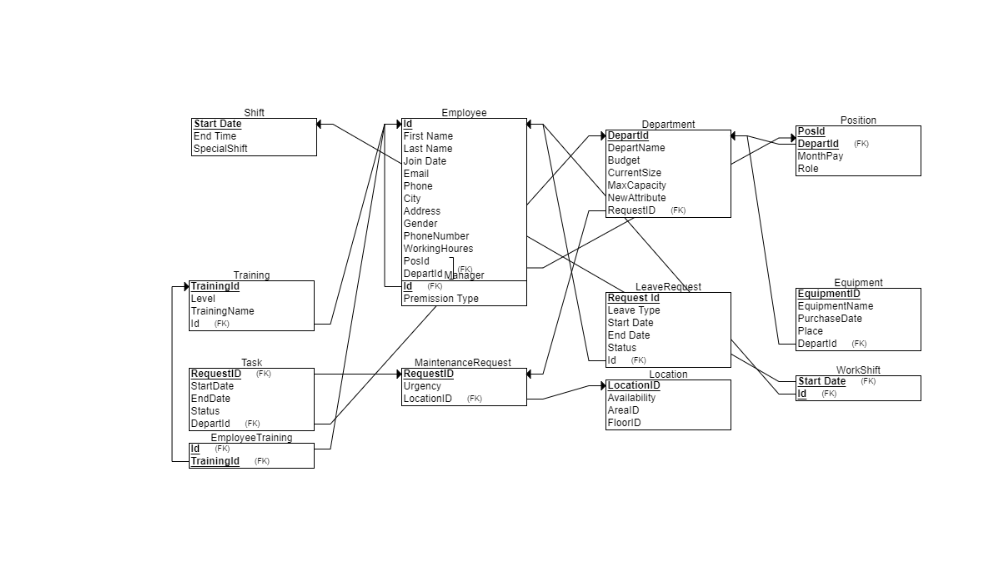
ניתן לראות את השדות והתיאורים והאילוצים שלהם.

ניקח את הסקריפט הזה ונעשה לו הינדוס לאחור (על ידי כלים שיש בתוכנה PL/SQL) ונוציא את דיאגרמת DSD של האגף שלהם – (כמובן שזה נמצא גם בתיקיה שלב ד' שעלתה לגיטהאב)  
A screenshot of a computer

Description automatically generated

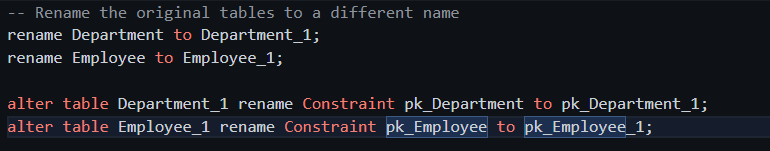
לאחר שיצרנו את הדיאגרמה אנחנו נעשה עוד שלב אחורה ונייצר את הERD של הבסיס הנתונים של האגף שלהם -   
  
ניקח את דיאגרמת ERD של האגף שלנו -   


לאחר שיש לנו את שתי האלה אנחנו יכולים לייצר את הERD המשותף של שתי האגפים -   


במהלך האיחוד בין שתי הדיאגרמות היינו צריכים לבצע כל מיני החלטות -  
1. באגף שלהם שדה "משכורת" היה בטבלת העובד, ואצלינו זה היה בטבלת "Position"- אצלהם השכר חושב לפי שעה ואצלינו זה היה שכר חודשי אז החלטנו להשאיר את שניהם ככה שלכל עובד יש שכר שעתי וגם יש לנו מידי על כמה שעתי זה יוצא...   
2. בבסיס הנתונים שלהם טבלת "מחלקה" מחוברת בקשר ישר לטבלת "עובד", אצלינו יש את טבלת ""position והחלטנו להשאיר את זה ככה כדי שבאיחוד שלהם יהיה גם את טבלת position  
3. בטבלת העובד שלהם הייתה תכונה של "תחילת עבודה" ואצלינו היה "תאריך כניסה" שזה בעצם אותו הדבר, אז החלטנו להשאיר רק אחד מהם.(כמובן שהכנסנו את הנתונים מ"תחילת עבודה" ל"תאריך כניסה" כמו שצריך.  
  
אחרי שעשינו את הדיאגרמת ERD נוציא ממנה את דיאגרמת DSD של האיחוד בין האגפים -   


עכשיו לאחר שיש לנו דיאגרמה של האיחוד בין האגפים, נוכל לייצר את הטבלאות המתאימות.

נשנה את השמות של הטבלאות ושמות המפתחות העיקריים –



לאחר מכן נוסיף את הטבלאות החדשות למערכת באופן זמני –

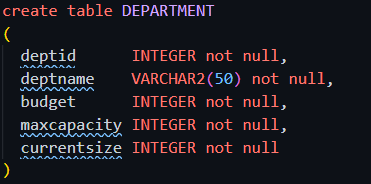
A black screen with a white text

Description automatically generated

נתבונן בטבלאות שקיימות בשתי המערכות ונשלים את ההבדלים הדרושים –

A black background with colorful text

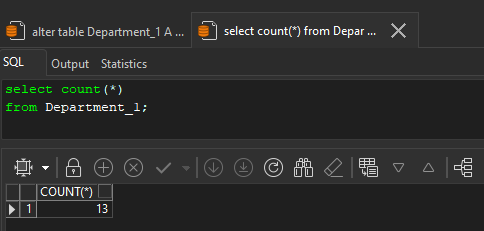
Description automatically generatedטבלה ראשונה: טבלת המחלקות –



נצטרך להוסיף שלוש שדות בטבלה שלנו (שמאל) ולכן נריץ את הפקודות הבאות –



וכעת אפשר להכניס את הערכים של הטבלה החדשה לטבלה הקודמת שלנו –

הטבלה לפני הערכים החדשים

A computer screen shot of a program

Description automatically generated

נכניס כעת את הערכים שאינם נמצאים בטבלה הישנה לתוך הטבלה החדשה

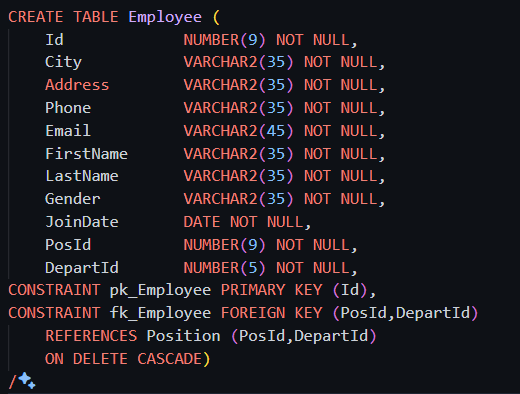
A screenshot of a computer

Description automatically generated

וכעת נראה את השינויים

נראה שנוספו עוד שורות לטבלה הישנה שלנו

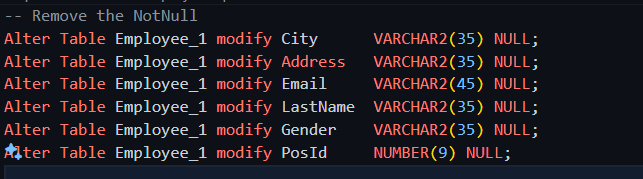
A computer screen shot of a program

Description automatically generatedטבלה שניה: טבלת העובדים –

מימין הטבלה שלנו ומשמאל הטבלה של המערכת החדשה

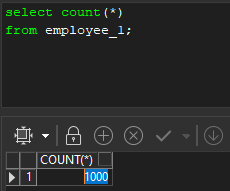
ניתן לראות שההבדלים הרבים הם בעיקר שמות שונים אבל התכונה אותה תכונה (JoinData ו-StartWork מהווים את אותו תכונה – זמן הצטרפות העובד לעבודה)

לכן נדאג כעת

* להוריד את האילוצים של Not Null על מנת שנוכל להכניס את הערכים בצורה תקינה
* להוסיף את השדות שלא קיימים

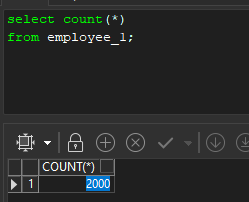
נוריד את האילוצים על העמודות האלו בגלל שהערכים האלה אינם קיימים בטבלה החדשה  
בנוסף את הערכים של השם המלא יכנס לשדה של השם הפרטי

נוסיף שורות שלא היו קיימות בטבלה הישנה. אין אילוץ כי השורות הקודמות אין להם את הערכים האלה

הטבלה לפני השינוי

A screenshot of a computer program

Description automatically generated נכניס כעת את הערכים לטבלה שלנו

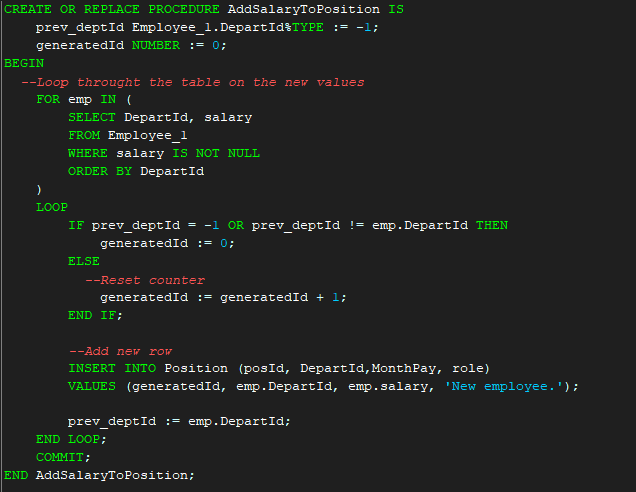
וכעת אנחנו רואים כי אכן שורות חדשות נוספו לטבלה שלנו שורות מהטבלה החדשה

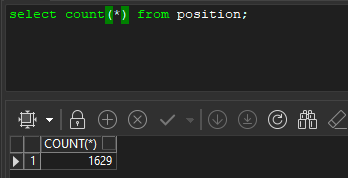
כעת המצב הוא שיש לנו את שדה המשכורת ב-2 טבלאות – נרצה לדאוג שהם יהיו רק בטבלת המשרות  
לכן נגדיר פרוצדורה שתיקח את השורות מהטבלה של העובדים בה הערך של המשכורת הוא לא נאל (כלומר השורות החדשות) ונדאג ליצור קוד חדש של PosId ועבור כל עובד אנחנו נקדם את הערך של המזהה  
עבור כל מחלקה חדשה נאתחל את הערך שנציב במזהה וכך לא נפגום ביחודיות של המפתחות בטבלה של המשרות  
בנוסף נגדיר את הRole עם ערך שרירותי היות ולא ניתן לנו שום ערך עבור שם המשרה

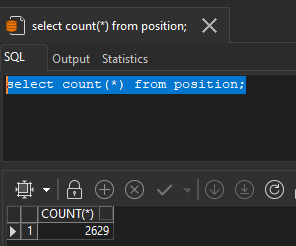


ונשנה את ההגדרות של הטבלה - נוסיף את השדה עבור תשלום חודשי כמו שהעובדים החדשים מקבלים  
ונוריד את האילוץ עבור תשלום שעתי עבור עובדים שאינם מקבלים את שכרם עבור שעה

וכעת נריץ את הפרוצדורה הבאה על מנת להכניס את ערכי התשלום החודשי עבור העובדים החדשים –



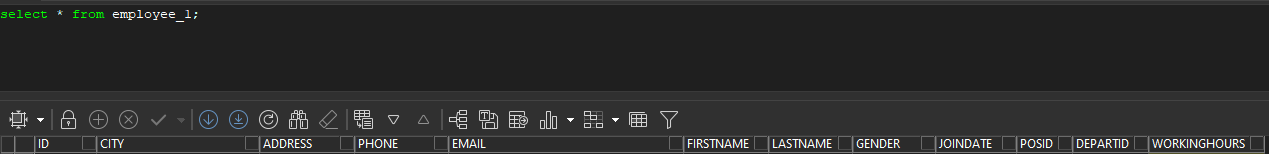
הערכים לפני הכנסת הנתונים

וכעת אחרי שהערכים הוכנסו על ידי הפרוצדורה

וכעת נוריד את שדה הsalary מטבלת העובדים –



וכעת הטבלה נראית כך –



וכעת כל מה שנותר הוא לזרוק את הטבלאות החדשות הכפולות, להחזיר את שמות המפתחות והאינדקסים

קודם נצטרך להגדיר מפתחות זרים של טבלאות אחרות שמצביעות לטבלאות שאנחנו עומדים לזרוק – מצאנו שעבור task יש מפתח זר כזה   
לכן ניצור מפתח זר חדש שמתייחס לdepartment\_1

-- Remove the old one

alter table task

drop CONSTRAINT fk\_Task\_Dept;

-- Create a new one for the updated table

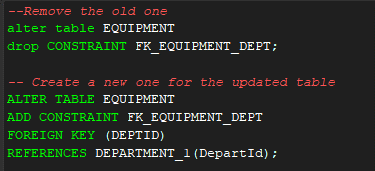
ALTER TABLE task

ADD CONSTRAINT fk\_Task\_Dept

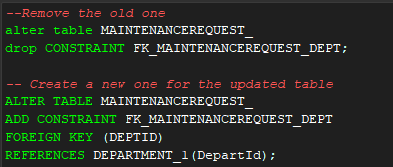
FOREIGN KEY (DEPTID)

REFERENCES DEPARTMENT\_1(DepartId);

וגם בטבלת הEQUIPMENT ולכן גם שם נמחק את המפתח זר ונגדיר אחד חדש –



וגם בטבלת הMAINTENANCEREQUEST\_ ונגדיר אחד חדש-

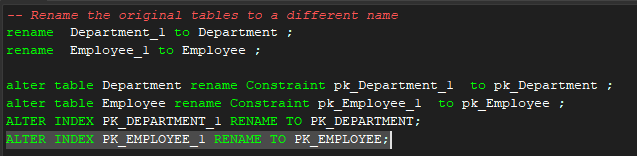
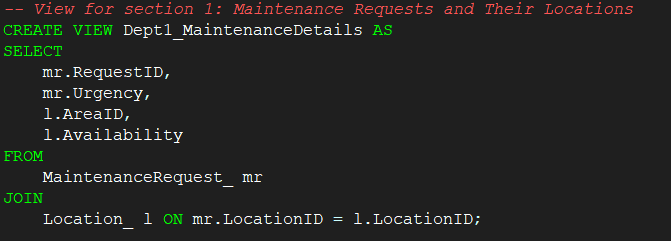
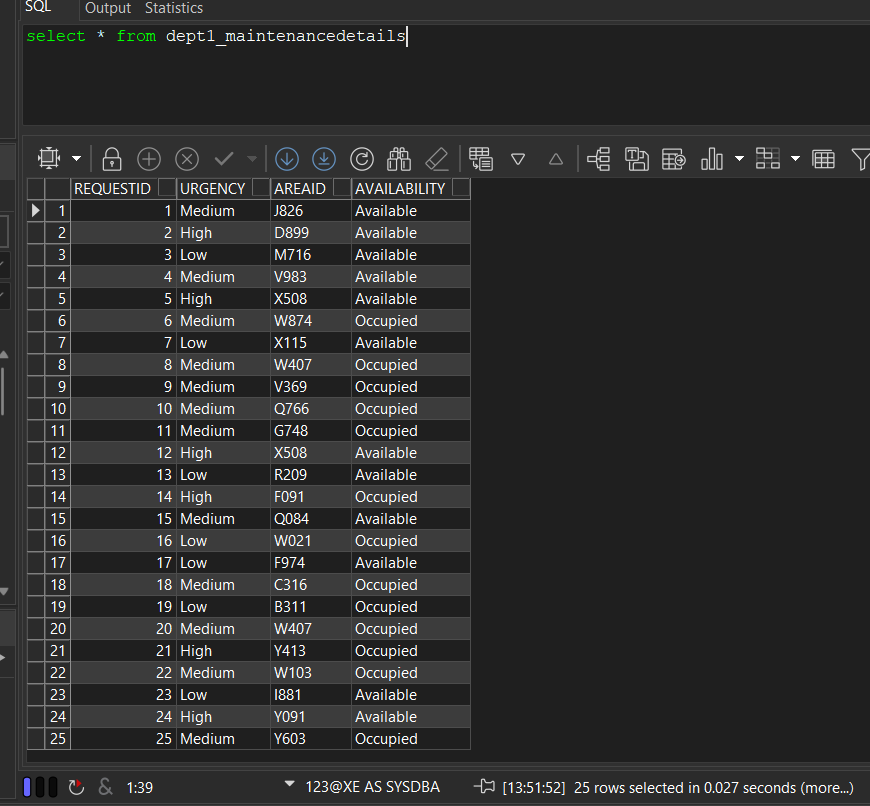
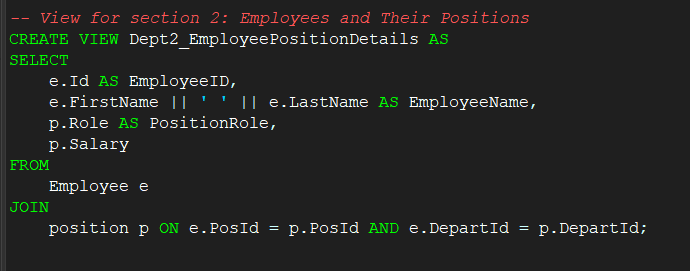


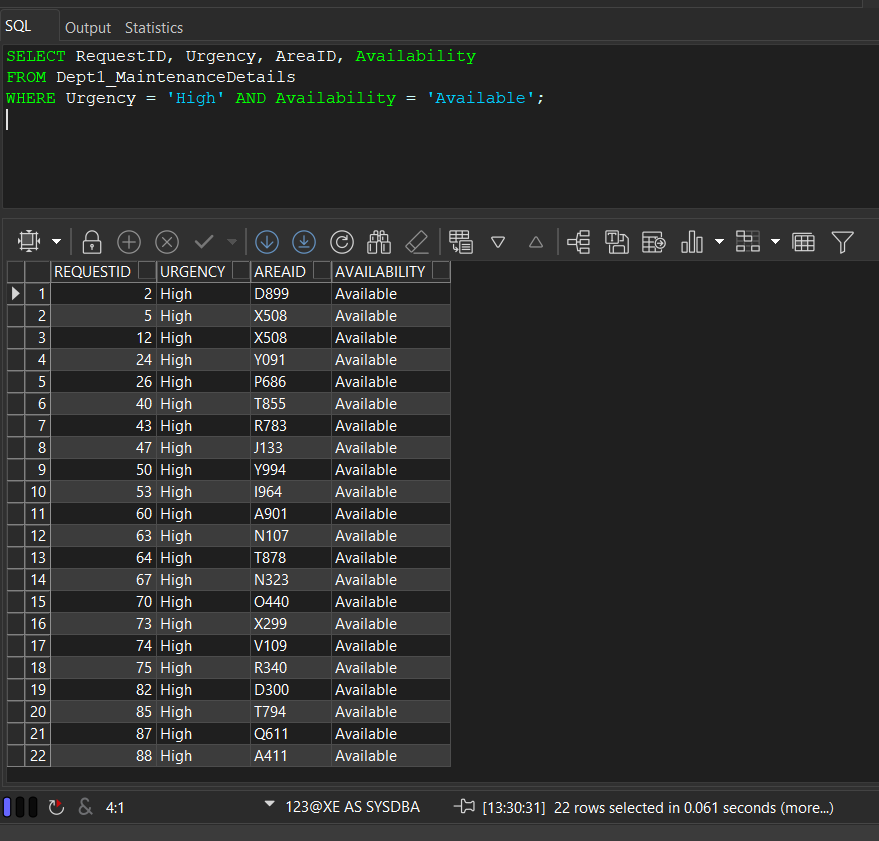
וכעת נוכל למחוק את הערכים של הטבלאות הישנים ולשנות את הטבלאות לשנו למה שהם היו קודם –



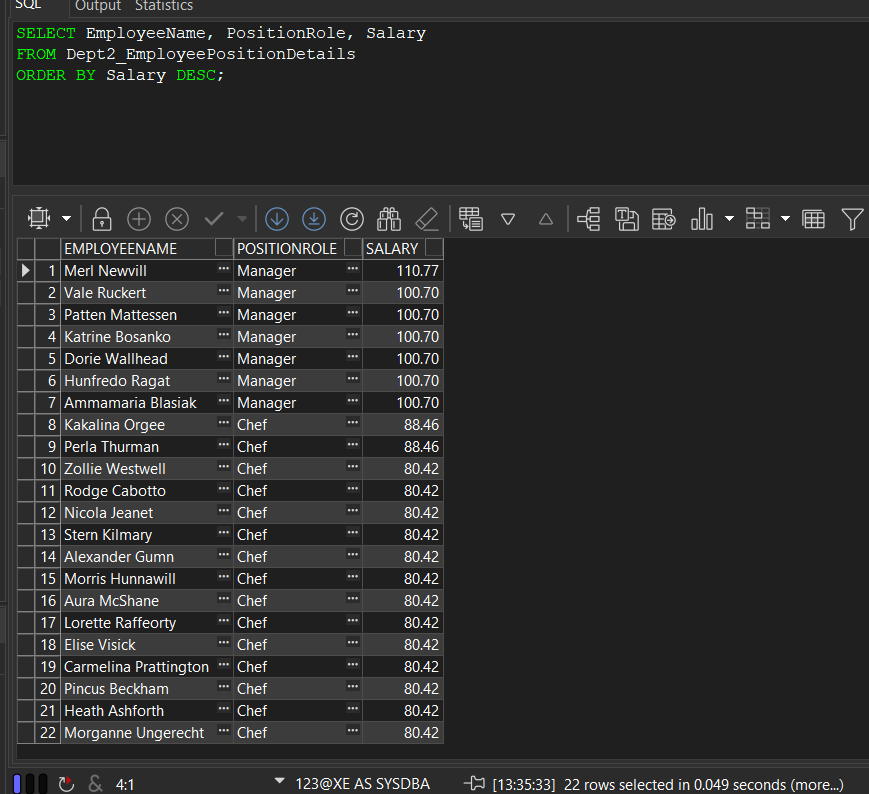


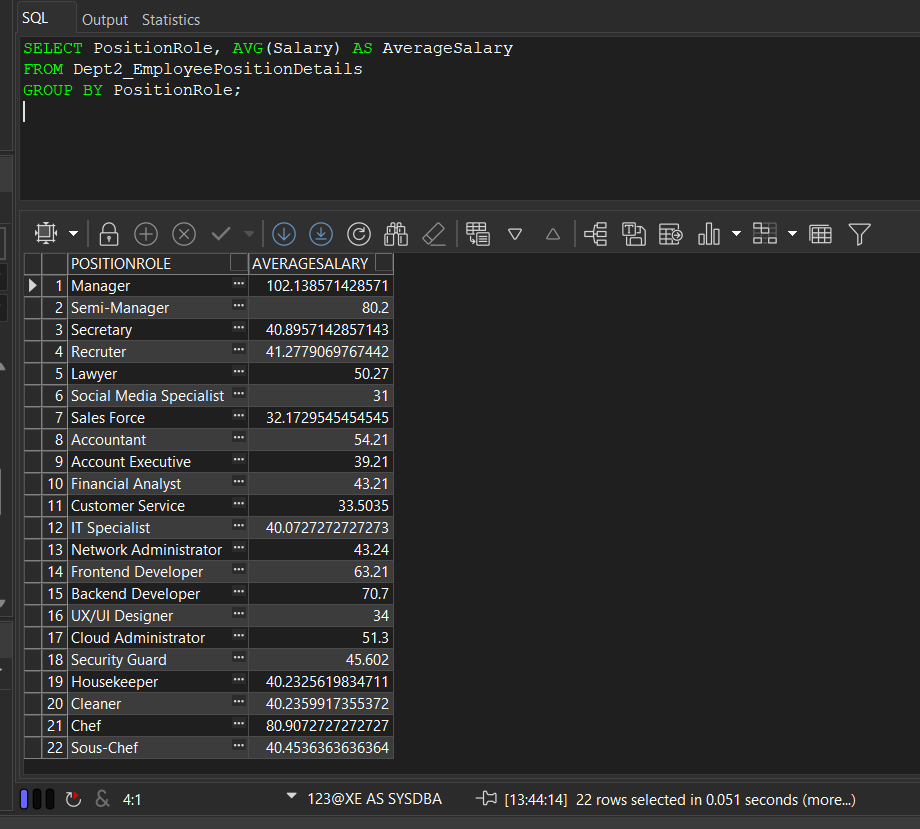
וכעת נשנה את שמות המפתחות הראשיים של הטבלאות שלנו למה שהם היו וגם את שמותם –

  
  
  
**Views -**בשלב זה אנחנו ניצור 2 מבטים שכל אחד מהם מציג תמונת מצב של אגף אחר מבין האגפים שאליהם עשינו את האינטגרציה.(הם נמצאים בגיטהאב שקובץ (VIEWS.SQL   
מבט מספר 1 -   
****מבט זה מכיל בקשות תחזוקה ופרטי מיקומים, כולל דחיפות וזמינות. הוא מועיל בכך שהוא מאפשר למנהלי תחזוקה לראות במהירות את בקשות התחזוקה הקריטיות ואת זמינות האזורים לטיפול. זה משקף את הצד של אגף 1.  
הנה דוגמית מהנתונים שהמבט מכיל -   
  
  
  
  
מבט מס' 2 -   
  
מבט זה מכיל פרטי עובדים ותפקידיהם, כולל משכורות. הוא מועיל בכך שהוא מספק למנהלי כוח אדם כלי לבחינת חלוקת תפקידים ושכר בארגון, מה שמסייע בקבלת החלטות על הקצאות שכר וקידומים. זה משקף את הצד של אגף 2.  
הנה דוגמית מהנתונים שהאגף מכיל -   
A screenshot of a computer

Description automatically generated  
 **שאילתות על המבטים -** שאילתה מס 1 על אגף מס' 1 -  
  
שאילתה זו מציגה את כל בקשות התחזוקה שהוגדרו כדחופות והאזורים הזמינים. היא מועילה לארגון בכך שהיא מאפשרת למנהלי תחזוקה לזהות במהירות את הבקשות הקריטיות והאזורים שניתן לטפל בהם מיד.

שאילתה מס' 2 על אגף מס' 1 -   
A screenshot of a computer

Description automatically generated  
שאילתה זו סופרת את מספר בקשות התחזוקה בכל אזור. היא מועילה לארגון בכך שהיא מספקת תמונה כללית על עומס התחזוקה בכל אזור, מה שמסייע בתכנון ובניהול משאבי התחזוקה בצורה יעילה יותר.  
  
  
  
  
  
  
  
שאילתה מס' 1 על אגף מס' 2 -   


שאילתה זו מציגה את שמות העובדים, תפקידיהם ומשכורותיהם, ומסדרת את התוצאות לפי המשכורת בסדר יורד. היא מועילה לארגון בכך שהיא מאפשרת למנהלים לראות את ההיררכיה של המשכורות בכל מחלקה, מה שמסייע בניתוח עלויות שכר.  
  
שאילתה מס' 2 על אגף מס' 2 -   


שאילתה זו מחשבת את ממוצע המשכורות לכל תפקיד. היא מועילה לארגון בכך שהיא מאפשרת לנתח את ממוצע המשכורות לפי תפקידים שונים, מה שיכול לעזור בקבלת החלטות לגבי מדיניות שכר והעלאות שכר.