**Gruppenmitglieder:**

1. Ramy Abdelhakim Elsaed Mohamed Ahmed (7215780)
2. Warnakulasooriya Christian Rodrigo (7206373)

**Gebuchter Anwendungskontext (**Verwaltung von Studentenwohnheimen und -unterkünften**):**

Unsere Gruppe hat sich für die Entwicklung einer Datenbankanwendung zur Verwaltung eines Studentenwohnheims entschieden. Die Anwendung soll es dem Verwaltungspersonal ermöglichen, effizient Informationen über Bewohner bzw. Studenten, Wohnung und Wohnungsheim Zuweisungen und weitere relevante Daten zu verwalten.

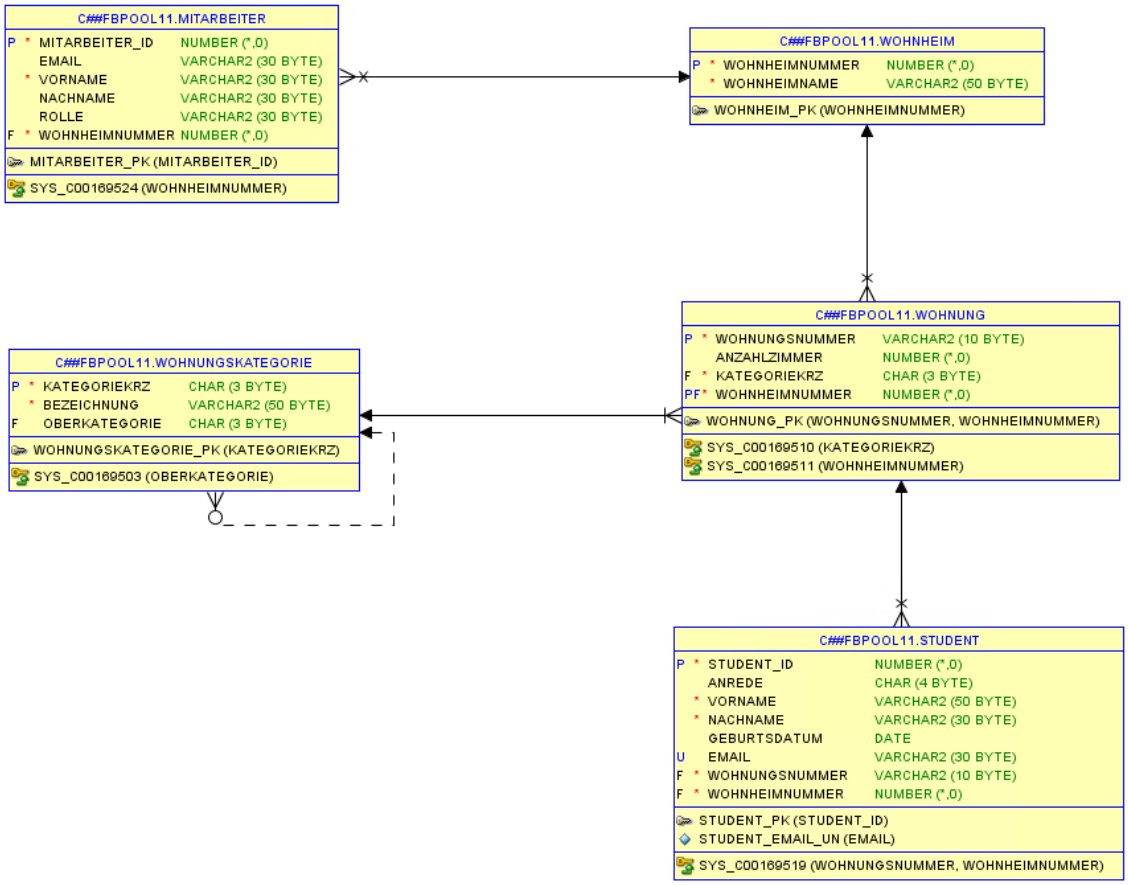
**Anwendungsszenario:**

Die Datenbankanwendung wird alle relevanten Informationen zu den Studenten und Wohnungen Studentenwohnheim speichern. Das Verwaltungspersonal kann neue Bewohner hinzufügen, Wohnung zuweisen. Jeder Bewohner wird mit seinem Namen, ID, Kontaktinformationen und weiteren persönlichen Details erfasst. Die Wohnung wird eindeutig identifiziert, und es wird erfasst, welcher Bewohner welche Wohnung bewohnt.

**Integritätsbedingungen:**

1. **ON DELETE CASCADE:**
    - Mit der Option ON DELETE CASCADE wird festgelegt, dass wenn der referenzierte Datensatz in der Elterntabelle gelöscht wird, alle zugehörigen Datensätze in der Kindertabelle automatisch gelöscht werden. Das bedeutet, dass die Löschaktion sich rekursiv auf die verknüpften Datensätze in der Kindertabelle auswirkt. Im Beispiel mit den Tabellen " Wohnung"," Student"," Mitarbeiter" wird das ON DELETE CASCADE angewendet.
2. **NOT NULL-Constraint:**
    - as NOT NULL-Constraint wird verwendet, um sicherzustellen, dass eine bestimmte Spalte in einer Tabelle keine NULL-Werte akzeptiert. In der Regel wird dies für Spalten verwendet, in denen NULL-Werte nicht zulässig sind und jede Zeile einen gültigen Wert für diese Spalte haben muss. Im Beispiel mit der Tabelle " Wohnheim " wird das NOT NULL-Constraint auf die Spalte " Wohnheimname " angewendet.
3. **Eindeutigkeits- oder UNIQUE-Integrität:**
    - Die E-Mail-Adresse jeder Student wird als eindeutige Information gespeichert. Dies gewährleistet, dass keine zwei Student die gleiche E-Mail-Adresse haben. Dadurch wird die Eindeutigkeit der Kontaktinformationen sichergestellt.

Diese Integritätsbedingungen sind entscheidend, um sicherzustellen, dass die Daten im System konsistent und korrekt bleiben, und tragen dazu bei, die Qualität der Verwaltung von Studentenwohnheimen zu verbessern.




Alle Tabellen verfügen über einen Primärschlüssel, der im Diagramm durch ein blaues "P" gekennzeichnet ist. Der Primärschlüssel dient als eindeutige Identifikationsnummer, beispielsweise für jedes Wohnheim, jeden Studenten oder Mitarbeiter. Ein roter Stern im Diagramm signalisiert, dass diese Attribute nicht den NULL-Wert annehmen dürfen.

Zum Beispiel, die Primärschlüssel der Tabelle "Wohnheim" werden von den Tabellen "Wohnung", "Mitarbeiter" und "Student" als Fremdschlüssel verwendet. Dies ist im Diagramm durch ein "F" markiert, das für "Foreign Key" steht. Die Verwendung von Fremdschlüsseln schafft Beziehungen zwischen den Tabellen, die durch Pfeile dargestellt werden. Es existieren verschiedene Pfeilarten, die die Kardinalitäten zwischen den Tabellen beschreiben. Im Folgenden erläutere ich die verschiedenen Kardinalitäten zwischen den Tabellen.

## Beziehung zwischen Wohnheim und Mitarbeiter

Ein Wohnheim kann mehrere Mitarbeiter haben, was durch die Crow's Foot Notation verdeutlicht wird. Jeder Mitarbeiter muss jedoch mindestens einem Wohnheim zugeordnet sein. Im Falle der Löschung eines Wohnheims werden auch alle Mitarbeiter gelöscht, die diesem Wohnheim zugeordnet waren. Dies wird durch das X neben der Crow's Foot-Notation symbolisiert, das auf eine "On Delete Cascade"-Constraint für den Fremdschlüssel hinweist. Dies bedeutet, dass eine "One or Many"-Beziehung zwischen beiden Tabellen besteht.

**Beziehung zwischen Wohnheim und Wohnung**

Die Beziehung zwischen dem "Wohnheim" und den "Wohnungen" ist ebenfalls als "One or Many" zu charakterisieren, da die Notation identisch mit der zuvor beschriebenen Beziehung ist. Ein Wohnheim kann demnach ein oder mehrere Wohnungen haben, wobei jede Wohnung genau einem Wohnheim zugeordnet ist. In der Tabelle "Wohnung" gibt es eine besondere Eigenschaft, nämlich einen zusammengesetzten Primärschlüssel. Die Kombination aus Wohnungsnummer und Wohnheimnummer fungiert als eindeutige Identifikation für jede Wohnung.

**Beziehung zwischen Wohnungskategorie und sich selbst (Rekursive Beziehung)**

In der Tabelle "Wohnungskategorie" besteht eine rekursive Beziehung, da der Fremdschlüssel "Oberkategorie" auf die Primärschlüssel "Kategoriekrz" verweist. Die Crow's Foot-Notation verdeutlicht, dass mehrere Wohnungskategorien einer Oberkategorie zugeordnet sein können. Der Kreis neben dem Crow's Foot zeigt an, dass die Oberkategorie einer Wohnungskategorie auf NULL gesetzt wird, falls die entsprechende Kategorie gelöscht wird. Dies bedeutet, dass eine "Zero or Many" Beziehung existiert.

**Beziehung zwischen Wohnung und Wohnungskategorie**

Die Beziehung zwischen den Tabellen "Wohnung" und "Wohnungskategorie" ist als "One or Many" gekennzeichnet, was im Diagramm durch das Crow's Foot-Symbol und einen Senkrechtstrich dargestellt wird. Dies bedeutet, dass eine Wohnungskategorie eine oder mehrere Wohnungen umfassen kann.

**Beziehung zwischen Wohnung und Student**

Die Beziehung zwischen den Tabellen "Wohnung" und "Student" spiegelt ebenfalls eine "One or Many"-Beziehung wider. Dies wird im Diagramm durch das Crow's Foot-Symbol symbolisiert. Somit bedeutet dies, dass ein oder mehrere Studenten in einer Wohnung wohnen können. Das X neben der Crow's Foot-Notation symbolisiert, das auf eine "On Delete Cascade"-Constraint für den Fremdschlüssel hinweist. Und wenn die Wohnung nicht mehr existiert bzw. gelöscht wird, wird es auch der entsprechende Student gelöscht.