**Aufgabe B) Logischer Datenbank-Entwurf**

**Relationale Umsetzung von Entity-Typ:**

**Legende:**  AK: Alternate Key (Alternativschlüssel)  
 FK: Foreign Key (Fremdschlüssel)  
 varchar(<x>): Zeichenkette mit max. X Zeichen  
 number: Stellvertretend für Datentypen int, float, double  
 <Attribute> NOT NULL: Attribut muss Wert haben (NULL-Werte nicht erlaubt)  
 IS NULL: Attribut darf den Wert NULL annehmen, ist demnach optional<Attribute> boolean: Wahrheitswert true/false  
 date: Datumsdatentyp in der Form dd.MM.yyyy hh:mm:ss

sch(Ferienwohnung)= ({[WohnungsID: number, Beschreibung: varchar, Preis/Tag: decimal, Größe (qm): number, Zimmeranzahl: number, AdressID: number]}, ∑Ferienwohnung)

∑Ferienwohnung: {(Beschreibung IS NULL), (Preis/Tag NOT NULL), (Preis/Tag > 0), (Größe(qm) NOT NULL), (Größe(qm) > 0), (Zimmeranzahl > 0)}

AK1: {AdressID}

FK: Ferienwohnung({AdressID}) -> Adresse ({AdressID})

sch(Zusatzausstattung) = {[Beschreibung: varchar]}

sch(Bilder) = ({[BID: number, Dateipfad: varchar, Beschreibung: varchar, WohnungsID: number]}, ∑Bilder)  
∑Bilder: {(Dateipfad NOT NULL), (Beschreibung NOT NULL)}

FK: Bilder({WohnungsID}) -> Ferienwohnung({WohnungsID})

sch(Adresse) = ({[AdressID:number, Strasse: varchar, HausNr: varchar, PLZ: varchar, OrtsID: number]}, ∑Adresse)

∑Adresse: {(Strasse NOT NULL), (PLZ NOT NULL), (HausNr NOT NULL)}

FK: Adresse ({OrtsID}) -> Ort ({OrtsID})

sch(Ort) = ({[OrtsID: number, Ortsname: varchar, ISO-Code: varchar, IATA-Code: varchar]}, ∑Ort)  
∑Ort: {(Ortsname NOT NULL, IATA-Code NOT NULL)}

FK: Ort ({ISO-Code}) -> Land ({ISO-Code})  
 FK: Ort ({IATA-Code}) -> Flughafen ({IATA-Code})

sch(Land) = ({[ISO-Code: varchar, Name: varchar]}, ∑Land)  
∑Land: {(Name NOT NULL)}  
AK = {Name}

sch(Touristenattraktion) = ({[AttraktionsID: number, Beschreibung: varchar, Name: varchar, AdressID: number]}, ∑Touristenattraktion)  
∑Touristenattraktion: {(Beschreibung NOT NULL), (Name NOT NULL)}

FK: Touristenattraktion({AdressID}) -> Adresse({AdressID})

sch(Flughafen) = ({(IATA-Code: varchar, Name: varchar)}, ∑Flughafen)  
∑Flughafen: {(Name NOT NULL)}  
AK = {Name}  
 FK: Flughafen ({Name}) -> Fluggesellschaft({Name})

sch(Fluggesellschaft)= ({[Name: varchar, Servicequalitätskennzahl: number]}, ∑Fluggesellschaft)  
∑Fluggesellschaft: {(0 < Servicequalitätskennzahl < 11)}

sch(Kunde) = ({[KundenNr: number, Name: varchar, Vorname: varchar, Geburtsdatum: date, Telefonnummer: varchar, E-Mail-Adresse: varchar, AdressID: number]}, ∑Kunde)  
∑Kunde: {(Name NOT NULL), (Vorname NOT NULL), (Geburtsdatum NOT NULL), (Telefonnummer NOT NULL), (E-Mail-Adresse NOT NULL)}  
AK1 = {E-Mail-Adresse}  
AK2 = {AdressID}

FK: Kunde ({AdressID}) -> Adresse ({AdressID})

sch(Bankverbindung) = ({[IBAN: varchar, KontoNr: number, BLZ: varchar, BIC: varchar, KundenNr: number]}, ∑Bankverbindung)  
∑Bankverbindung: {(KontoNr NOT NULL), (BLZ NOT NULL), (BIC NOT NULL)}  
AK1 = {Konto-Nr, BLZ}  
AK2 = {Konto-Nr, BIC}

FK: Bankverbindung ({KundenNr}) -> Kunde ({KundenNr})

sch(Belegung) = ({(BelegungsID: number, Status-Flag: varchar(12), von: date, bis: date, Belegungsdatum: date, KundenNr: number, WohnungsID: number]}, ∑Belegung)  
∑Belegung: {(Status- Flag NOT NULL), (von NOT NULL), (bis NOT NULL), (Belegdatum NOT NULL), (KundenNr NOT NULL), (Status-Flag ∈ {‘Reservierung‘, ‘Buchung‘})

FK: Belegung ({KundenNr}) -> Kunde ({KundenNr})

FK: Belegung({WohnungsID}) -> Ferienwohnung({WohnungsID})

sch(Rechnung) = ({[RechnungsNr: number, Zahlungseingangsdatum: date, Rechnungsdatum: date, Betrag: decimal, BelegungsID: number]}, ∑Rechnung)

∑Rechnung: {((Rechnungsdatum NOT NULL), (Betrag NOT NULL)}

AK1 = {BelegungsID}

FK: Rechnung ({BelegungsID}) -> Belegung ({BelegungsID})

sch(entfernt von) = ({[Startort: number, Zielort: number, km:number]}, ∑entfernt von)  
∑entfernt von = {FK: entfernt von({Startort}) -> Ort ({OrtsID})  
 FK: entfernt von({Zielort}) -> Ort ({OrtsID})}

sch(fliegt an) = ({[Start: number, Ziel: number, FGS: varchar]}, ∑fliegt an)

∑fliegt an = {FK: fliegt an({Start}) -> Flughafen ({IATA-Code})  
 FK: fliegt an({Ziel}) -> Flughafen ({IATA-Code})  
FK: fliegt an({FGS}) -> Fluggesellschaft({Name})

sch(besitzen) = ({[Beschreibung, WohnungsID]}, ∑besitzen)  
∑besitzen = {FK: besitzen({Beschreibung}) -> Zusatzausstattung({Beschreibung})  
 FK: besitzen({WohnungsID}) -> Ferienwohnung({WohungsID})}

**Inter-relationale Integritätsbedingungen** ∑FW**:**

∑FW: {(∀ f ∈ Flughafen ∀ fl ∈ Fluggesellschaften: f.Name ≠ fl.Name)}

∑FW: {(∀ fw ∈ Ferienwohnung ∀ z ∈ Zusatzausstattung:  
fw.Beschreibung ≠ z.Beschreibung)}

∑FW: {(∀ fw ∈ Ferienwohnung ∀ b ∈ Bild: fw.Beschreibung ≠ b.Beschreibung)}

∑FW: {(∀ so ∈ Startort ∀ zo ∈ Zielort: so.Ortsname ≠ zo.Ortsname)}

∑FW: {(∀ be ∈ Belegung ∀ r ∈ Rechnung: be.Belegunsdatum >= r.Rechnungsdatum)}

∑FW: {(∀ sf ∈ Startflughafen ∀ zf ∈ Zielflughafen: sf.IATA-CODE ≠ zf.IATA-Code)}

∑FW: {({(∀ o ∈ Ort: Distanz > 0)}