Das CASE-Tool *Data Architect* unterstützt die relationale Datenmodellierung auf der konzeptionellen und physischen Ebene (Relationenmodell) und generiert aus dem physischen Modell ein SQL-DDL-Skript zur Generierung der Datenbankstruktur. Hierbei werden die wichtigsten auf dem Markt vertretenen DBMS mit unterschiedlichen Release-Ständen unterstützt, in der Version 9.0 sind dies 75 unterschiedliche Systeme.

Das Tool arbeitet bzgl. relationaler DBMS hauptsächlich in zwei Modi, dem *Conceptual Data Model (*.cdm)* und dem *Physical Data Model (*.pdm),* aus dem dann ein *SQL-Skript (*.sql)* generiert werden kann. Darüber hinaus besteht die Möglichkeit, die Grafiken der Datenmodelle ganz oder ausschnittsweise als **.emf*, **.bmp*, **.jpg* etc. zu extrahieren.

Neben dem Aufbau von Datenmodellen (Analyse und Design) unterstützt das Tool auch das *Re-Engineering von relationalen Datenbanken*, indem entweder eine Verbindung zum DB-Server hergestellt wird, sodass das Tool die Daten aus dem Systemkatalog auslesen kann, oder indem man dem Tool das DDL-Skript zur Generierung der DB-Struktur zur Verfügung stellt.

CONCEPTUAL DATA MODEL (*.CDM) Ebene der konzeptionellen Datenmodellierung **ANALYSE**	**PHYSICAL DATA MODEL (*.PDM)** Ebene der physischen (relationalen) Datenmodellierung **DESIGN**	**SQL-DDL-SKRIPT (*.SQL)** **IMPLEMENTIERUNG**
Die wichtigsten grafischen Komponenten der Toolbar zur konzeptionellen Datenmodellierung sind - **Entity** - **Relationship** - **Inheritance** (Supertyp-/Subtyp-Hierarchie) Jede der grafischen Komponenten kann nach Markierung und Doppelklick auf Registerkarten detailliert beschrieben und bearbeitet werden: - Entities erhalten **Attribute,** ggf. als Identifier (primary key), - Relationships werden über ihre **Kardinalitäten** und **Abhängigkeiten** näher beschrieben (identifizierend = dependent), bei - Inheritance-Beziehungen kann man unterschiedliche **Optionen zur physischen Implementierung und zur Vererbung** wählen. **Achtung:** Auf dieser Ebene werden noch keine physischen Referenzen in Form von Foreign-Key-Spalten angegeben!	Das Tool generiert aus dem konzeptionellen Modell ein physisches Datenmodell im Hinblick auf ein ausgewähltes DBMS (Datentypen, SQL-„Dialekte“ etc.). Die wichtigsten grafischen Komponenten, die ggf. mit Hilfe der Toolbar ergänzt werden können: - **Table** - **Reference**. Auch hier können die grafischen Komponenten wieder nach Markierung und Doppelklick auf Registerkarten bearbeitet werden: - Tables enthalten **Columns (Spalten)**, denen **Primary-Key-** und / oder **Foreign-Key**-Constraints als Implementierung der Relationships des konzeptionellen Modells zugeordnet sind.	Nach einem **syntaktischen Check**, der auch beim Übergang vom konzeptionellen zum physischen Modell gemacht wird, steht das SQL-Skript im Editor zur Ansicht / Überarbeitung zur Verfügung. Jeder Check muss mit **„0 errors“** und sollte mit **„0 warnings“** abgeschlossen werden – alle warnings, die vom User akzeptiert werden, sollten vom User auch verstanden worden sein! Bei der Generierung werden die Datentypen und SQL-„Dialekte“ des ausgewählten DBMS berücksichtigt.
Übergang vom konzeptionellen zum physischen Modell: **Tools / Generate Physical Model ... / <DBMS wählen>** Es empfiehlt sich zunächst, alle voreingestellten Optionen zu verwenden.	Übergang vom physischen Modell zum SQL-Skript: **Database / Generate Database ... / Script generation** Es empfiehlt sich zunächst, alle voreingestellten Optionen zu verwenden.	In einer Message-Box wird unter dem Hinweis „Usage“ beschrieben, wie das SQL-Skript im DBMS importiert werden kann.