

MWC3 '25 Database Design Problem Statement

As students, you may be aware that several FAFSA issues delayed financial aid processing and awarding this past year. Fortunately, your school heard about your database expertise and recruited you to design their new database system to help keep track of financial aid and scholarship awards.

Your task is to design a relational database to store the data needed in creating, reading, updating and deleting financial aid scholarships to students. The financial aid office began storing scholarship data in Excel, as shown in Figures 1, 2, and 3 (see the end of this document) for the first 13 records. Each record contains 18 attributes which are described below. Each record represents 1 award to 1 student. Assuming an average of 20,000 students attending your school each semester, an average of 3 awards per student per semester, and 2 semesters per academic year, the office quickly decided that an Excel spreadsheet would not work long-term (~120,000 records per year). Use standard SQL commands.

Deliverables

You are required to submit the following documents:

1. 1 or more images showing your ER diagram.
2. 1 SQL file with the queries for the table creation, insertion of the 13 records into the database, and reading the result set containing all awards (student id, fund names and award amounts) for student 16895.

Attribute Descriptions

1. **StudentId**: An integer that uniquely identifies each student
2. **ExternalId**: An integer that uniquely identifies each student in the national FAFSA database
3. **HSGraduation**: The student's high school graduation date
4. **State**: The student's legal state of residence
5. **Legal Residence**: The student's month and year they became legal residents in the state above. The format is YYYYMM, where Y is a year digit and M is a month digit (e.g. 202401 is January, 2024)
6. **ParentsState**: The parent's legal state of residence
7. **ParentsResidence**: The parent's month and year they became legal residents in the state above. The format is YYYYMM, where Y is a year digit and M is a month digit (e.g. 202401 is January, 2024)
8. **EFC**: Expected Financial Contribution. This is a dollar amount (rounded to the nearest dollar) that FAFSA expects the family to contribute towards the educational expenses.
9. **FundId**: An integer uniquely identifying a scholarship fund.
10. **Fundname**: The short code name assigned to a scholarship fund.
11. **AwardId**: An integer uniquely identifying an award given to a student
12. **AwardStatusCode**: A code representing the award status. Allowed status codes are {ACCEPTED, DECLINED, PENDING, CANCELLED UNDER REVIEW}
13. **Amount**: The dollar amount (rounded to the nearest dollar) awarded to the student from the fund

14. **AcceptedAmount:** The dollar amount (rounded to the nearest dollar) accepted by the student from the fund
15. **ModifiedBy:** A string (either system or the username of the person) representing who last updated the record
16. **ModifiedOn:** The time (format HH:MM:SS) the record was last modified. Because of batch processing, the system can modified many records within a second.
17. **COAfirstdollaraward:** A dollar amount (rounded to the nearest dollar) for the cost of attendance for the semester. This attribute includes tuition, room and board, book, etc.
18. **COAlastdollaraward:** A dollar amount (rounded to the nearest dollar) for the cost of attendance for the semester. This attribute includes tuition only.

Fig. 1. First 6 student attributes for the first 13 records

	A	B	C	D	E	F	G
1	StudentId	ExternalId	HSGraduation	State	LegalResidence	ParentsState	ParentsResidence
2	16798	3152	6/1/2024	MI	200604	MI	199510
3	16798	3152	6/1/2024	MI	200604	MI	199510
4	16798	3152	6/1/2024	MI	200604	MI	199510
5	16798	3152	6/1/2024	MI	200604	MI	199510
6	16873	3434	1/1/2024	MI	200605	MI	197212
7	16873	3434	1/1/2024	MI	200605	MI	197212
8	16873	3434	1/1/2024	MI	200605	MI	197212
9	16873	3434	1/1/2024	MI	200605	MI	197212
10	16895	3504	6/1/2024	MI	200510	MI	197004
11	16895	3504	6/1/2024	MI	200510	MI	197004
12	16911	3547	5/1/2024	MI	200504	MI	197510
13	16911	3547	5/1/2024	MI	200504	MI	197510
14	16911	3547	5/1/2024	MI	200504	MI	197510

Fig. 2. Second 6 student attributes for the first 13 records

	A	H	I	J	K	L	M
1	StudentId	EFC	FundId	FundName	AwardId	AwardStatusCode	Amount
2	16798	25645	1009	LEDP	43766	ACCEPTED	2000
3	16798	25645	1398	MG13	43767	ACCEPTED	5000
4	16798	25645	1381	MI1A	43852	ACCEPTED	2500
5	16798	25645	1380	MI2A	43851	ACCEPTED	3000
6	16873	1958	1406	CMTA	44997	ACCEPTED	75
7	16873	1958	1381	MI1A	33744	ACCEPTED	2500
8	16873	1958	1380	MI2A	33745	ACCEPTED	3000
9	16873	1958	2	PELA	24515	ACCEPTED	5435
10	16895	14739	980	CENS	44248	ACCEPTED	14310
11	16895	14739	986	CFES	44249	ACCEPTED	450
12	16911	23362	1398	MG13	24517	ACCEPTED	5000
13	16911	23362	1381	MI1A	33746	ACCEPTED	2500
14	16911	23362	1380	MI2A	33747	ACCEPTED	3000

Fig. 3. Last 5 student attributes for the first 13 records

	A	N	O	P	Q	R
1	StudentId	AcceptedAmount	ModifiedBy	ModifiedOn	COAfirstdollaraward	COAlastdollaraward
2	16798	2000	system	5:31:43	32105	14760
3	16798	5000	system	5:31:43	32105	14760
4	16798	2500	system	5:31:43	32105	14760
5	16798	3000	system	5:31:43	32105	14760
6	16873	75	system	9:37:08	32105	14760
7	16873	2500	system	9:37:05	32105	14760
8	16873	3000	system	9:37:05	32105	14760
9	16873	5435	system	9:37:05	32105	14760
10	16895	14310	system	5:31:43	32105	14760
11	16895	450	system	5:31:43	32105	14760
12	16911	5000	system	5:31:43	32105	14760
13	16911	2500	system	5:31:43	32105	14760
14	16911	3000	system	5:31:43	32105	14760

MWC3 '25 Database Design Grading Rubric

<i>Item</i>	<i>Excellent</i>	<i>Proficient</i>	<i>Functional</i>	<i>Needs Improvement</i>	<i>Possible Points</i>
Database Design and Structure	15: Database schema is highly organized with clear and efficient relationships between tables. It normalizes the data (at least 3NF) effectively.	10: Schema is well-structured with appropriate relationships, and most data is normalized (at least 2NF).	5: Schema has some redundancies, and data is only partially normalized. Relationships between tables are somewhat unclear.	0: Schema is poorly structured with many redundancies, and relationships are unclear or absent. Data normalization is not addressed.	15
Table Creation	10: All tables (students, awards, scholarships, etc.) are properly created with appropriate constraints (primary keys, foreign keys, etc.).	8: Tables are created with primary and foreign keys, but some constraints may be missing or inconsistent.	5: Tables are created but missing key constraints, leading to potential issues with data integrity.	0: Tables are poorly created or missing necessary constraints (e.g., no primary or foreign keys).	10
Data Types and Field Choices	20: Excellent choice of data types for each attribute, optimizing for storage and performance while ensuring data integrity.	15: Appropriate data types are selected for most fields, though some may be suboptimal or not fully standardized.	10: Data types are generally correct but may lead to inefficiencies or potential data integrity issues.	0: Data types are poorly chosen, leading to inefficient storage or potential errors in data handling.	20

Relationships (ER Model)	15: The ER diagram is comprehensive, including all entities (students, awards, scholarships, etc.) and their relationships, with clear foreign keys.	10: ER diagram is mostly complete but may lack clarity in some relationships or entities.	5: ER diagram is incomplete or contains some inaccurate or unclear relationships between entities.	0: ER diagram is incomplete or significantly flawed in representing relationships between entities.	15
Data Entry	20: INSERT statements are correctly written and efficient. Data is entered without issues, and the system can handle large datasets easily.	15: INSERT statements are mostly correct, though some inefficiencies or issues may exist with bulk data entry.	10: Data entry works, but INSERT statements may be prone to errors or inefficient when handling large datasets.	0: Data entry is problematic due to incorrect or inefficient INSERT statements.	20
Query Functionality (CRUD Operations)	10: CRUD operations (Create, Read, Update, Delete) are implemented efficiently using optimized SQL queries that work seamlessly across tables.	7: CRUD operations are implemented well, with good SQL queries for most tasks, though some queries may need optimization.	5: CRUD operations function but are not optimized or could lead to performance issues with large datasets.	0: CRUD operations are incomplete or inefficient, with poorly written SQL queries or missing functionalities.	10